

**Коммерциялық
емес акционерлік
қоғам**



**Алматы
энергетика және
байланыс
университеті**

Компьютерлік
технологиялар
кафедрасы

.....

ORACLE ДЕРЕКТЕР ҚОРЫН БАСҚАРУ ЖҮЙЕЛЕРІ

1 бөлім

5B070400 - Есептеу техникасы және бағдарламалық қамтамасыз ету мамандығының студенттері үшін зертханалық жұмыстарды орындау бойынша әдістемелік нұсқаулықтар

Алматы 2014

Құрастырушылар: Аманбаев А.А., Сатимова Е.Г. ORACLE деректер қорын басқару жүйелері. 1 бөлім: 5B070400 - Есептеу техникасы және бағдарламалық қамтамасыз ету мамандығының студенттері үшін зертханалық жұмыстарды орындау бойынша әдістемелік нұсқаулықтар. - Алматы: АЭЖБУ, 2014. – 78 б.

Әдістемелік көрсетелімде ДҚБЖ Oracle мысалында заманауи мәліметтер базасын құрудың негізі бойынша зертханалық жұмыстарды жүргізуді қамтамасыз ететін материалдар қамтылған. Оқу құралы мәліметтер базасын басқаруға және SQL сұрауларының тіліне арналған. Материалда баяндалады: базалық сұраулардың сипаттамаларынан бастап, біріктірулер, ішкі сұраулар мен транзакциялар арқылы жасалатын күрделі манипуляциялармен аяқталады. Аталмыш оқу құралын жасаудың негізгі мақсаты «ORACLE мәліметтер базасын басқару жүйелері» пәнін оқитын «Есептеу техникалары және бағдарламалық қамсыздандыру» мамандығының студенттері үшін ДҚБЖ Oracle бағдарламасының ерекшеліктерін есепке ала отырып, SQL стандартында командаларды құрудың машықтарын бекіту керектігі болып табылады.

Без. - 15, әдебиет көрсеткіші – 16 атау.

Пікір беруші: аға оқытушы Альмуратова К.Б.

«Алматы энергетика және байланыс университеті» коммерциялық емес акционерлік қоғамының 2014 жылға арналған басылым жоспары бойынша шығарылады.

© «Алматы энергетика және байланыс университеті», ҚЕАҚ, 2014 ж.

Мазмұны

Кіріспе.....	4
1 Зертханалық жұмыс №1. ORACLE сұрауларымен жұмыс істеуге арналған құралдар.....	5
1.1 Зертханалық жұмысқа тапсырмалар.....	5
1.2 Зертханалық жұмысты орындауға арналған әдістемелік нұсқаулар	5
2 Зертханалық жұмыс №2. ДҚБЖ ORACLE қолданушылардың қауіпсіздігін басқару.....	10
2.1 Зертханалық жұмысқа тапсырмалар.....	10
2.2 Зертханалық жұмысты орындауға арналған әдістемелік нұсқаулар	12
3 Зертханалық жұмыс № 3. ДҚБЖ ORACLE деректер қорларының объектілерін құру.....	18
3.1 Зертханалық жұмысқа тапсырмалар.....	18
3.2 Зертханалық жұмысты орындауға арналған әдістемелік нұсқаулар	20
4 Зертханалық жұмыс № 4. SQL ДҚБЖ ORACL-нің арнайы операторлары мен функциялары.....	29
4.1 Зертханалық жұмысқа тапсырмалар.....	29
4.2 Зертханалық жұмысты орындауға арналған әдістемелік нұсқаулар	31
5 Зертханалық жұмыс № 5. SQL ДҚБЖ ORACLE-нің күрделі операторлары және көпжолдық функциялары	36
5.1 Зертханалық жұмысқа тапсырмалар.....	37
5.2 Зертханалық жұмысты орындауға арналған әдістемелік нұсқаулар	38
6 Зертханалық жұмыс №6. Oracle функцияларының қосымша мүмкіндіктері.....	40
6.1 Зертханалық жұмысқа тапсырмалар.....	40
6.2 Зертханалық жұмысты орындауға арналған әдістемелік нұсқаулар	41
Әдебиеттер тізімі.....	44

Кіріспе

Аталмыш әдістемелік нұсқаудың құрамында ДҚБЖ Oracle мысалында заманауи деректер қорын құрудың негізі бойынша зертханалық жұмыстарды жүргізуді қамтамасыз ететін материалдар қамтылған. Оларда ДҚБЖ Oracle қарағанда SQL процедуралық емес тілінің ерекшеліктері қарастырылады.

Зертханалық жұмыстардың негізгі мақсаты деректер қорын басқару, SQL тіліндегі сұраулар және олармен жұмыс істеу туралы тәжірибелік машықтарды игеру.

Зертханалық жұмыстар 3 бағыт бойынша жинақталған - ДҚБЖ Oracle администрацияларына бөлу (минималды бөлігі), SQL тілінің өзін жанама игеру (1 бөлім) және базалық тілдерден SQL – PL/SQL (2 бөлім) тілін процедуралық түрде кеңейтіп конструкциясын және құрылымын зерттеу. Олар ДҚБЖ Oracle қалай жұмыс істейтінің түсінуге мүмкіндік береді. Материалдар Oracle Education Student Kit «Oracle Database 11g: Administration Workshop I» [1]. «Oracle Database 11g: SQL Fundamentals I - II» және «Oracle Database 11g: PL/SQL Fundamentals» [2] курстарын тыңдаушылардың басшылығына негізделген.

Зертханалық жұмыстардың деректері «Деректер қорын жобалау» курсының логикалық жалғасы болып табылады, ол жерде студенттер SQL тілі мен оның жүзеге асуы – Transact SQL туралы білім алған болатын. Оларда ДҚБЖ Oracle қарағанда SQL процедуралық емес тілінің ерекшеліктері қарастырылады.

Зертханалық жұмыс жұмыс тапсырмасын орындаудан, оқытушының тапсырмасынан, есепті безендіруден және жасалған жұмысты қорғаудан тұрады.

Жұмыс тапсырмалары қарастырылып отырған тақырып бойынша қандай да бір тапсырманы орындау бойынша нақты жұмыстарды қамтиды.

Әрбір зертханалық жұмысты орындау есепті безендірумен аяқталуы тиіс, оның құрамында мыналар болады:

- Жұмыстың мақсаты мен тапсырмасы.
- Теориялық дайындық туралы қысқаша қорытынды.
- Жасалған жұмыстың нәтижелері (жиналған тәжірибе мен сынақтардың, өңделген деректердің және т.б. алынған нәтижелері).
- Жұмыс бойынша қорытындылар және пайдаланылған әдебиеттердің тізімі.

Орындалған жұмыстар мен есепті безендіру оқытушының алдында қорғалады.

Зертханалық жұмыс курсық жобаның сәйкес бөлігін орындау шартында ғана есептеледі.

Есеп электрондық түрде безендірілуі мүмкін.

1 Зертханалық жұмыс №1. ORACLE сұрауларымен жұмыс істеуге арналған құралдар

Жұмыстың мақсаты: Oracle - SQL*Plus, Toad, Oracle SQL Developer сұрауларын жазу үшін интегралданған ортаны пайдалануды үйрену.

1.1 Зертханалық жұмысқа тапсырмалар

1. Sqlplus қолданып, Oracle серверіне қосылу.
2. Зертханалық жұмыста келтірілген мысалдарды орындау.
3. Oracle SQL Developer пайдаланып, Oracle серверіне қосылу.
4. HR сұлбасында EMPL кестесін құру.
5. Toad пайдалану арқылы Oracle серверіне қосылу.
6. HR сұлбасын қарау және ER-диаграммасын тұрғызу.

Бақылау сұрақтары: Oracle сұрауларын жазу үшін өндеудің интегралданған ортасын атап өтіңіз.

1.2 Зертханалық жұмысты орындауға арналған әдістемелік нұсқаулар

Жұмысты орындау реті.

Берілген зертханалық жұмыс ДҚБЖ Oracle-де интегралданған құру орталарының көмегімен дерек қордағы ақпараттарды өңдеу үшін сұраныстарды құру әдістерін үйретеді.

Зертханалық жұмыстар қашықтағы (виртуалды машинаға орнатылған) Oracle серверіне берілген құралдардың көмегімен қосылу арқылы орындау қарастырылған.

Жиі қолданатын құралдар:

- 1) Quest Software фирмсының Toad.
- 2) Осы фирманың SQL Navigator.
- 3) Allround Automations фирмсының PL/SQL Developer.
- 4) Oracle фирмсының Oracle SQL Developer.

SQL*Plus – ЖДҚ серверлерімен жұмыс жасауға мүмкіндік беретін Oracle құралы [3]. SQL*Plus өзінің жеке командалар тіліне ие:

- 1) Пакетті және интерактивті сұрауларды орындау.
- 2) SQL командалары мен PL/SQL блоктарын енгізу, редактрлеу, орындау, сақтау, таңдау мен сақтау.
- 3) Сұрауларды форматтау, есептеу, сақтау және нәтижелерін шығару.
- 4) Кез келген кестенің бағаналарының анықтамаларын қарау.
- 5) Соңғы пайдаланушыларға хабарлама жіберу және жауаптар қабылдау.
- 6) Деректер қорын администрациялау бойынша жұмыстар орындау.

SQL операторын үш әдіспен аяқтауға болады (нүктелі үтір (;), бос жолдан тек қана қисық сызықтың (/) көмегі арқылы және бос жол арқылы).

Oracle SQL Developer. Oracle SQL Developer-де деректер қорының кестесіндегі ақпараттарды өндеудің мүмкіндіктерін жүзеге асруға келесі артықшылықтарды қолдануға болады:

- 1) Ол ресми түрде тегін және Oracle 11g арқылы қойылады.
- 2) Ол Oracle-дің өзінде жасап шығарылған, сондықтан кәсіпорындарда бекітілген бағдарламалардың тізіміне түскен кезде мәселелер туындамайды.
- 3) Ол Java тілінде жазылған, сондықтан бір ғана грфикалық интерфейсті Windows-та да, Unix-те де қолдана беруге болады.
- 4) Оны сондый-ақ, басқа ДҚБЖ-лерге сұрау салу үшін қолдануға болады, мысалы, Microsoft SQL Server және Microsoft Access, сондықтан кәсіпорынның алуан түрлі деректер қорына сұрауларды жазу үшін бір ортаны пайдалану мүмкіндігі пайда болады.

Оның кемшіліктері де бар: Oracle тарапынан қолдау қызметінің ресми түрде болмауы, ресурсымдылығы, жұмыстағы кейбір тұрақсыздықтар. Oracle SQL Developer салыстырмалы түрде тікелей пайда болды және сондықтан да кәсіпорындарда әлі кең көлемде тарала қойған жоқ, бірақ аталмыш курста SQL Developer және Toad қолданылатын болады.

ER-үлгілеудің визуалды үлгілеу мүмкіндігі бар, деректер сұлбасын генерациялайтын және деректер қорының реверс-инженерингі бар SQL Developer кеңейтілген версиясы жеке өнім ретінде SQL Developer Data Modeler деген атаумен қарастырылады (ол да тегін).

Oracle SQL Developer жұмыс кезінде бірінші істелуге тиісті — деректер көзімен жаңа байланыстар құру керектігі. Мұны File | New мәзірінен де, Connection терезесінен де жасауға болады (Connections конвейтеріне тышқанның оң жақ батырмасын шерту керек және контексті мәзірден New батырмасын шертеміз).

Жаңа байланыстың параметрлерін енгізгеннен кейін барлық параметрлер дұрыс екеніне көз жеткізу үшін бұл терезеде Test батырмасын қолдануға болады. Осыдан кейін қосылу параметрлерін келесі қолданыс үшін сақтауға болады (Save батырмасы) және серверге кіруді орындау керек (Connect батырмасы). SQL кодының редактрлеу терезесі ашылады, ол жерден SQL және PL/SQL үшін код таңдауға және теруге болады (ол ресми түрде SQL Worksheet деп аталады).

TOAD зерттеу құралы. TOAD аспаптық құралы (Tool for Oracle Application Development) Oracle серверлік қосымшасын зерттеудің әлдеқайда атақты құралдарының бірі болып табылады.

TOAD іске қосуы керекті пиктограммаға тышқанды екі рет шерту арқылы жүзеге асады. Сервер қосылғаннан кейін экранда диалог терезесі пайда болады, ол жерде қолданушы Oracle сервері арқылы ертеректе орнатылған байланыстардың бірін таңдай алады немесе Database, User/Schema және Password жолақтарына жаңа байланыстың параметрлерін енгізе алады.

Сервермен байланысқаннан кейін экранда бағдарламаның басты терезесі ашылады, ол жерде қолданушы керекті аспаптық құралдарды қабылдап алады. TOAD жұмыс үдерісі кезінде ДҚ-мен бірнеше байланыстар орнатуға мүмкіндік береді. Байланыстарды басқару үшін келесідей мәзір пункттері пайдаланылады:

File New Connection – Oracle серверімен жаңа байланыс орнатуға мүмкіндік береді;

File End Connection – Oracle-мен ағымдағы байланысты жабады.

File End All Connection – Oracle-мен барлық байланыстарды жабады.

File New Connection – Oracle серверімен жаңа байланыс орнатуға мүмкіндік береді;

File End Connection – Oracle-мен ағымдағы байланысты жабады.

File End All Connection – Oracle-мен барлық байланыстарды жабады.

Жұмыс үдерісі кезінде келесі пункттер қолжетімді болады:

Database Schema Browser – браузер терезесін ашады, оның көмегімен ағымдағы байланыста көрінетін сұлба нысандары туралы ақпараттарға жол ашады;

Database SQL Editor – SQL командалары мен скриптерін жазу және орындау үшін қолданылады;

Database Procedure Editor – PL/SQL блоктарымен жұмыс жасау үшін пайдаланушыға аспаптық құралдарды ұсынады;

Database SQL Modeller – SQL-сұрауларды визуалды құрастыру үдерісін қолдауды;

Database Export ... – пайдаланушыға ДҚ-да SQL скриптері түрінде деректерді түсіру мүмкіндігін береді;

Database Import ... – ДҚ-да деректерді импорттайды;

Database Commit – ағымдағы байланыста транзакцияның фиксациясын жүзеге асырады;

Database Rollback – ағымдағы байланысты транзакцияның шегінуі жүзеге асырылады.

*SQL*Plus арқылы серверге қосылу.* Қашықтағы деректер қорына екі әдіспен қосылауға болады:

1) CONNECT командасының көмегімен SQL*Plus арқылы.

2) SQL*Plus қосылған кезде командалық жолдар опциясының көмегімен.

Екі жағдайда да міндетті түрде қосылуы тиіс желілік қызмет арылы берілетін байланыс жолын көрсету керек. Байланыс жолының форматы машинадағы Oracle желілік қызметінің версиясына байланысты, сол арқылы қосылу жүзеге асырылады. Net8 пайдаланған кезде, байланыс жолдары *\$.ORACLE_HOME/network/admin/* каталогында орналапқан *tnsnames.ora* (8.1.x және одан да жоғары версияларында) файлындағы желілік қызметінің атына сәйкес келеді.

а) cmd командалық жолды қосыңыз.

b) Командалар жолы арқылы *sqlplus sys/password@orcl AS SYSDBA* командасының көмегімен *SQL*Plus* шақырыңыз.

c) *SQL> ALTER USER hr IDENTIFIED BY hr.*

SQL> ALTER USER hr ACCOUNT UNLOCK командаларын қолдана отырып *HR* қолданушысын бұғаттан шығарыңыз.

d) *SQL> CONN[ECT] hr/hr@orcl* командасы арқылы базаға қосылыңыз;

e) *SQL> SHOW USER* серверіне қандай қолданушы қосылғанын көріңіз.

Сценария файлын құрыңыз.

a) *SQL*Plus* сеансында әрбір қадамды ашқан сайын сценарий орындалатын *login.sql* файлыын құрыңыз. Бұл файлға *SERVEROUTPUT* ұқсас параметр беріңіз, ол автоматты түрде орындалатын болады. *login.sql* сценарийінің мысалы төменде көрсетілген (оны қолданылатын ортаға байланысты қайта өңдеуге болады).

```
DEFINE _EDITOR=vi  
SET serveroutput on size 1000000  
SET trimspool on  
SET long 5000  
SET linesize 100  
SET pagesize 9999  
COLUMN plan_plus_exp format a80
```

b) *Scott* деректер қорының сұлбасына қосылыңыз, *EMP* кестесінің сипаттамасын енгізіңіз, командалардың тізбегі орындалған жағдайда *script.sql* файлын қосыңыз (кестені скриптпен жасаңыз және кодты *script.sql* файлында жазыңыз). Осы командалардың орындалу нәтижелері *screen.lst* файлында көрінеді. Осы әрекеттер орындалғаннан кейін *SQL*Plus* утилиты арқылы шығып кетіңіз.

```
CONNECT scott/tiger@orcl;  
DESC EMP;  
SPOOL screen.lst;  
START sctipt.sql;  
SPOOL OFF;  
EXIT;
```

c) скрипт орындауға автоматты түрде жіберіңіз *C:\sctipt.sql sqlplus scott/tiger@orcl @C:\sctipt.sql*

Скрипттің орындалу нәтижелері командалық жолға тікелей енгізілетін болады.

d) Файлдағы нәтижелерді дискке енгізу үшін операциялық жүйені қайта бағыттау құралын пайдаланыңыз:


```
sqlplus -S scott/tiger@orcl @C:\Oracle.sql > C:\OracleResults.txt
```

Сервермен жұмыс жасауға арналған командалар.

a) Экрандағы файлға шығару ағынын сақтау:

```
SQL> SPOOL толық_жолы_берілген_файл_аты
```

b) Файлда шығару ағынан сақтауды аяқтау:

```
SQL> SPOOL OFF
```

c) Кестенің құрылымын қарау:

```
SQL> DESC кесте_аты
```

d) Кестенің мазмұнын қарау:

```
SQL> SELECT * FROM кесте_аты
```

e) Пайдаланушының кестесінің тізімін қарау:

```
SQL> SELECT table_name FROM user_tables
```

f) Пайдаланушының ұсыныстарының тізімін қарау:

```
SQL> SELECT view_name FROM user_views
```

g) Барлық ұсыныстардың тізімін қарау (соның ішінде деректердің сөздік-анықтамалық ұсыныстары да бар):

```
SQL> SELECT view_name FROM dba_views
```

h) Ағымдағы датаны алу (sysdate функциясы):

```
SQL> SELECT sysdate FROM dual
```

i) Кез келген функцияны қосу:

```
SQL> SELECT < функция_аты(параметрлері)> FROM DUAL
```

j) Кез келген процедураны қосу:

```
SQL> EXEC < процедура_аты (параметрлері)>
```

Команданы құру, үндеу жасау және айнымалыға байланысты шығару.

Келесі кодты орындаңыз

```
SQL> VARIABLE var1 CHAR(10)
```

```
SQL> BEGIN
```

```
  :var1:='Oracle';
```

```
  END;
```

```
  /
```

```
  PL/SQL процедурасы сәтті аяқталды. SQL> PRINT var1
```

```
VARI
```

Oracle SQL Developer арқылы серверге қосылу:

a) Connections қыстырмасындағы «+» басы арқылы серверге қосылыңыз.

b) hr сұлбасынан employees және departments кестелерінің негізінде автоматты түрде генерирасиалау арқылы құрылған сұранысты шығарыңыз және орындалған нәтижелерді Excel-ге экспорттаңыз.

```
SELECT employees.first_name, employees.last_name,
```

```
       departments.department_name FROM employees, departments
```

```
WHERE departments.department_id = employees.department_id;
```

с) Берілген тапсырманы шешу үшін келесі қимылдарды жасаңыз:

1) Oracle SQL Developer қосыңыз. File мәзірінен New таңдаңыз, содан кейін New Gallery терезесінен Database Connection таңдаңыз және ОК батырмасын басыңыз. Жаңа байланыс құрау терезесі ашылады.

2) Осы терезеге байланыс параметрлерін енгізіңіз және байланыс орнағанын тексеру үшін Test батырмасын басыңыз. Осыдан кейін Connect батырмасын басыңыз.

3) Connections терезесінде жасалған байланыс нысанының үстінен тышқанның оң жағын шертіңіз және контексті мәзірден Open SQL Worksheet таңдаңыз. SQL кодының редактор терезесі ашылады.

4) Осы терезеге тышқанның оң жағын шертіңіз. Query Builder сұрауларын құрушыларының графикалық ортасы ашылады.

5) Тізіменен employees и departments кестелерінің сол жақ бөлігіне алып келіңіз, ал содан кейін employee_id/manager_id артық байланыстарын алып тастаңыз. Шығарылған сұрау автоматты орналастырылатын редактор кодының терезесіне қайта оралу үшін пайдалану батырмасын басыңыз.

6) Сұруды орындау үшін F9 батырмасын шертіңіз. Содан соң нәтижелер терезесіне көрсеткіш қойыңыз да, Select мәзірінен SelectAll таңдаңыз. Содан соң нәтижелер терезесіндегі бөлінген облысқа тышқанның сол жағын шертсеңіз және контексті мәзірден ExportData | XLS таңдаңыз. XLS сақталған файлының атын енгізіңіз, пайдалану батырмасын басыңыз, ал содан кейін экспорт сәтті өткеніне көз жеткізу үшін өзіңіз жасаған Excel-дегі XLS файлын ашыңыз.

2 Зертханалық жұмыс №2. ДҚБЖ ORACLE қолданушылардың қауіпсіздігін басқару

Жұмыстың мақсаты: ORACLE құралдары арқылы қолданушыларды, сұлбалар мен кестелерді және олардың элементтерін құру, өңдеу және жоюды үйрену.

2.1 Зертханалық жұмысқа тапсырмалар

Командалық жолда мына мүмкіндіктер сұлбасы жүзеге асырыла алатын скрипт құру: *A қолданушысы* және *E қолданушысы* *HR.REGIONS* кестесінде *INSERT* құқығына ие болсын. *A қолданушысы* *B қолданушысына* *HR.REGIONS* кестесі үшін *INSERT* мүмкіндігі берілсін, оның үстіне *WITH GRANT OPTION* көрсетілімімен бірге (1 кезең). *B қолданушысы* бұл мүмкіндікті *C қолданушысына* берсін (2 кезең). Содан соң *C қолданушысы* дәл осыны *E қолданушысынан* да алсын (3 кезең). Әрі қарай *C қолданушысы* айтып өтілген мүмкіндікті *D қолданушысына* берсін (4 кезең). *A қолданушысы* *INSERT* мүмкіндігін *B қолданушысы* үшін алса, онда ол *C қолданушысы* ретінде де белгілене алмайды, себебі ол оны ертеректе *E қолданушысынан* алып қойған болатын. Егер де *E қолданушысы* берілген

мүмкіндікті *C* қолданушысына апарып берген болса, онда *B* қолданушысының мүмкіндігін жою келесі *C* және *D* қолданушылары үшін мүмкіндіктердің каскадты жойылу артықшылығы берілуі тиіс (2.1 кесте қараңыз).

2.1 кесте

А қолданушысы	В қолданушысы	С қолданушысы	Д қолданушысы	Е қолданушысы
GRANT INSERT ON HR.REGIONS TO B WITH GRANT OPTION	Құқық алу			
	GRANT INSERT ON HR.REGIONS TO C WITH GRANT OPTION	В-дан құқық алу. Е- ден құқық алу.		GRANT INSERT ON HR.REGIONS TO C WITH GRANT OPTION
		GRANT INSERT ON HR.REGIONS TO D	Құқық алу	
REVOKE INSERT ON HR.REGIONS FROM B	Құқықтың күшін жою	Құқықты сақтау	Құқықты сақтау	Құқықты сақтау

1. *stud* рөлін құру және осы рөлге екі қолданушыға *user1* және *user2* осы рөлді қосу. Рөлдің құқығын *stud* алып бару және тікелей *user2* қолданушысына.

stud рөлі үшін *HR.REGIONS* кестесінде көшіріп қою құқығын тоқтату.

2. Oracle балық қолданушылары үшін мүмкіндік және құқық мысалында түсіндіру.

3. *DEPARTMENTS* кестесі үшін синоним жасаңыз.

4. Синонимдерді қолдана отырып, *DEPARTMENTS* кестесіндегі барлық жолдарды сұраңыз.

Қолжетімді кестелер туралы ақпарат алу үшін *USER_TABLES* деректер сөздігін сұраңыз. Кіруге мүмкіндік ала алатын барлық кестелер туралы ақпараттарды көру үшін *ALL_TABLES* деректер сөздігінде ұсыныс сұраңыз. Сіз өзіңіз жасаған кестені оларға қоспаңыз.

Бақылау сұрақтары:

- 1) Пайдаланышының ресурстарына шектеуді зерттеңіз.
- 2) Пайдаланышының пароліндегі шектеуді зерттеңіз.
- 3) Кестелер құру үшін пайдаланушыға қандай мүмкіндіктер беру керек?
- 4) Егер сіз кесте құратын болсаңыз, кім сіздің кестеңіз үшін басқа пайдаланушыларға мүмкіндік бере алады?

5) Сіз – DBA. Сіз дәл осындай жүйелік мүмкіндіктерді талап ететін көптеген қолданушыларды жасайсыз. Нені қолданған жөн?

б) Өзіңіздің құпия сөзіңізді өзгерту үшін сіз қандай команданы пайдаланасыз?

2.2 Зертханалық жұмысты орындауға арналған әдістемелік нұсқаулар

Жұмысты орындау реті.

1. Пайдаланушылардың аккаунттын басқару:

a) *cmd* командалық жолағын жіберіңіз;

b) *sys* пайдаланушысынан *sqldeveloper-ge* кіріңіз;

Пайдаланушы құрыңыз:

```
CREATE USER test_user IDENTIFIED BY qwerty;
```

c) бұғаулау арқылы пайдаланушының күйін өзгертіңіз

```
ALTER USER test_user ACCOUNT LOCK;
```

d) немесе бұғаулаудан босатыңыз

```
ALTER USER test_user ACCOUNT UNLOCK;
```

e) пайдаланушы үшін кестелік кеңістікке квоталар белгілеңіз *test_user*:

```
ALTER USER test_user QUOTA 100M ON SYSTEM;
```

```
ALTER USER test_user QUOTA UNLIMITED ON USERS;
```

```
ALTER USER test_user QUOTA UNLIMITED ON EXAMPLE;
```

f) кестелік кеңістікте және пайдаланушының кестелік кеңістіктігінде квоталар туралы деректер алу үшін:

```
SELECT USERNAME,  
       DEFAULT_TABLESPACE,  
       TEMPORARY_TABLESPACE
```

```
FROM DBA_USERS
```

```
WHERE USERNAME = 'TEST';
```

g) пайдаланушыны деректер қорынан жойыңыз

```
DROP USER test_user CASCADE;
```

SYSTEM пайдаланушы арқылы *manager* құпия сөзімен жүйеге кіріңіз, ал содан кейін әрекет етіп отырған пайдаланушыны өзгертіңіз.

```
SQL> CONNECT SCOTT/TIGER@ORCL
```

2. Администраторлық SQL-сұрауларды орындау:

a) Oracle ДҚ-ның версиясы туралы деректер алыңыз

```
SELECT * FROM V$VERSION;
```

b) сервердегі ағымдағы дата мен уақытты алыңыз

```
SELECT TO_CHAR(sysdate,'dd/mm/yyyy hh24:mi:ss')  
FROM DUAL;
```

c) ДҚ-дағы барлық кестелердің аттарын алыңыз

```
SELECT TABLE_NAME  
FROM SYS.ALL_TABLES;
```

d) деректер қорындағы пайдаланушылардың аттарын алыңыз

```
SELECT USERNAME FROM USERS;
```

- е) пайдаланушының барлық объектілерін алу (кестелер, ұсыныстар, тізбектіліктер, индекстер және т.б.)

```
SELECT OBJECT_NAME  
FROM ALL_OBJECTS  
WHERE OWNER='...';
```

3. Пайдаланушының құқықтары мен рөлдері:

- а) *QWERTY* жүйесінде *RUSTEM* пайдаланушысын құпия сөзімен бірге құрыңыз

```
SQL> CREATE USER rustem IDENTIFIED BY qwerty  
2 DEFAULT TABLESPACE USERS  
3 TEMPORARY TABLESPACE TEMP  
4 /
```

- б) Пайдаланушыға құқықтар қатарын тағайындаңыз, бірінші дәрежелі – бұл сессияны серверде жасау. Ең бастысы, соңғы оператордың *COMMIT* жұмыс жасауы

```
GRANT CREATE SESSION TO RUSTEM  
/  
GRANT CREATE TABLE TO RUSTEM  
/  
GRANT CREATE PROCEDURE TO RUSTEM  
/
```

- с) профиль жасаңыз және оны біздің пайдаланушыға бекітіңіз:

```
CREATE PROFILE DUMMY_PROF LIMIT  
PASSWORD_LIFE_TIME 180;  
/
```

Сіз жасаған профиль парольдің жарамдылық мерзімін 180 күнге дейін шектейді. Осы профильді *DUMMY* пайдаланушысы үшін анықтаңыз:

```
SQL> ALTER USER DUMMY PROFILE TO DUMMY_PROF;
```

- д) Кейбір шектеулерімен бірге *clerk* профилін құрыңыз. Барлық көрсетілмеген профильдер *DEFAULT* профилінен алынады.

```
CREATE PROFILE clerk LIMIT  
SESSIONS_PER_USER 2  
CPU_PER_SESSION unlimited  
CPU_PER_CALL 6000  
LOGICAL_READS_PER_SESSION unlimited  
LOGICAL_READS_PER_CALL 100  
IDLE_TIME 30  
CONNECT_TIME 480;
```

- е) *DMROLE* бос рөлін жасаңыз:

```
SQL> CREATE ROLE DMROL
```

Оған *GRANT* операторының көмегімен мүмкіндіктер қатарын жасаңыз, мысалы *ALTER SESSION*:

```
SQL> GRANT ALTER SESSION TO DMROLE
```

Осы рөлді *GRANT* операторының көмегімен *DUMBO* пайдаланушысына енгізіңіз:

SQL> GRANT DMROLE TO DUMBO

h) Пайдаланушы мен оның сұлбасындағы барлық объектілерін жойыңыз:

SQL> DROP USER DUMBO CASCADE

Администрациялық есетік жазбалар және мүмкіндіктер. Администрациялық есетік жазбалар және мүмкіндіктер мынадай административтік функцияларды орындауға мүмкіндік береді: қолданушыларды басқару, деректер қорының жадымен басқару, сондай-ақ деректер қорын өсу және тоқтату.

SYS және SYSTEM қолданушылары. Әрбір деректер қорында автоматты түрде DBA рөлі белгіленген екі есептік жазбаның аты беріледі:

1) *SYS* (бастапқы пароль: CHANGE_ON_INSTALL).

2) *SYSTEM* (бастапқы пароль: MANAGER).

Олар екеуі де сіздерге инсталляцияда (Windows операциялық жүйесі) көрсетілген пароль арқылы немесе конфигурациялау арқылы (Linux операциялық жүйесі) құрылады.

1) *SYSTEM* – бұл деректер қорын қосу мен өшіруден басқа барлық администрациялық функцияларды орындау үшін сіз қолданатын пайдаланушының есептік жазбасы.

2) *SYS* - *SYS* сұлбасында орналастырылатын деректер қорының сөздігінің көрсетелімі мен базалық кестелері. Бұл базалық кестелер мен көрсетілімдер *Oracle Database XE* функциялауы үшін қажет. Деректер сөздігін бүтіндікпен қамтамасыз ету үшін *SYS* сұлбасындағы кестелермен тек деректер қорының серверін ғана басқарады. Олар ешқашан деректер қорының кез келген пайдаланушысы немесе администраторы тарапынан монифицирленбеуі керек. *Сіз SYS жүйесінде қандай да бір кестелер құру керек емес.*

Әдетте, *SYS* пайдаланушысына кіретіндей себеп болмайды. Қосу мен тоқтатудан басқа барлық администрациялық тапсырмаларды орындау үшін *SYSTEM* пайдаланушыларын қолдану ұсынылады.

SYSDBA жүйелік мүмкіндігі. Oracle деректер сөздігін ANY (мысалы, SELECT ANY TABLE) типіндегі жүйелік мүмкіндіктерінен қолданушыларды қорғауды ұсынады, осылайша сіздің деректер базаңыздың сөздігіне кіруге тек администрациялық мүмкіндіктері бар қолданушылар ғана мүмкіндік алады, яғни «AS SYSDBA» сияқты деректер қорына қосылу мүмкіндігі бар.

SYSDBA – бұл *SYS* қолданушылары үшін ғана арналып жасалған жүйелік мүмкіндік. Ол *SYS* қолданушысына жоғары деңгейлі администрациялық тапсырмаларды, деректер қорын қосу мен тоқтату секілді тапсырмаларды орындауға мүмкіндік береді. Қалыпты жағдайларда *SYS* қолданушысы атынан деректер қорына кіру керектігі тумаса да, *SQL* (*SQL*Plus*) командалық жолағына *SYS* арқылы қосылғыңыз келсе, онда сізге «AS SYSDBA» опцияларын міндетті түрде пайдалану керек болады. AS SYSDBA байланысы кезінде сіз SYSDBA мүмкіндігін аласыз. Егер сіз *SYS*

қолданушысы ретінде қосылған кезде AS SYSDBA опциясын жіберетін болсақ, онда SQL командалық жолағы қосылуды қайтарып тастайды.

SYSDBA мүмкіндігінің мүмкіндіктеріне шолу:

- 1) STARTUP және SHUTDOWN операцияларын орындау;
- 2) ALTER DATABASE: open, mount, back up, or change character set;
- 3) CREATE DATABASE, DROP DATABASE, CREATE SPFILE;
- 4) ALTER DATABASE ARCHIVELOG;
- 5) ALTER DATABASE RECOVER;
- 6) RESTRICTED SESSION мүмкіндігін қосады;
- 7) ALTER DATABASE RECOVER;
- 8) SYS қолданушы ретінде қосылуға мүмкіндік береді.

ORACLE ішкі және сыртқы аутентификацияларды пайдаланады.

Сыртқы аутентификация кезінде администрациялық құқықтары бар қолданушы келесідей әдіспен қосылады:

```
SQL > connect sys/пароль@строка_коннекта as sysdba
```

Келесі мысал SQL командтылық жолағынан SYSDBA мүмкіндіктерінің деректер базасымен байланыстырудың басшылығын жасайды:

```
SQL > connect sys/password as sysdba
```

password – бұл SYS есептік жазбасының пайдаланушылық құпия сөзі.

Пайдаланушылық аккаунтты басқару. Пайдаланушының аты. ДҚБЖ негізгі міндеттерінің бірі – бұл ақпаратты қарауға, құру мен модификациялауға осы үшін заңды себептері бар тұлғаларға ғана құқық беру мүмкіндігі. Бұл үшін пайдаланушылар идентификацияны табу керек және соның негізінде нақты анықталған нысандарға ғана қол жетімді болу керек.

Идентификацияның процедурасына дәстүрлі көзқарас мұнда жатыр: әрбір пайдаланушының өзінің қайталанбас пайдаланушы аты (*логин*) мен *құпия сөзі болады*. Сондымен қатар, ДҚБЖ Oracle әр түрлі альтернативті технологияларды пайдалануды қолдайды, мысалы смарт-карталар немесе құпия сөзімен үйлесетін басқа да физикалық құрылғылар, олар ортақ Oracle Advanced Security Option құралының опциялары арқылы PIN-кодтармен бірлесіп жұмыс жасайды.

Деректер қорында есептік жазба физикалық құрылым болып танылмайды. Оның үстіне, мәліметтер қорының барлық нысандары пайдаланушыға тиесілі болады. Пайдаланушының аты деректер қорының шегінде жалғыз болуы мүмкін. Оның ұзақтығы 30 символдан аспауы керек. Ат латын әріптерінен, цифрларынан, доллар белгісінен (\$) және астын сызу белгісімен (_) берілуі мүмкін. Пайдаланушының аты осы уақытқа дейін пайдаланылып қойған сөздер бола алмайды. Егер пайдаланушының атын қандай да бір арнайы символмен беру керек болса, онда мұны атты екі тырнақшаға алу арқылы жазу арқылы істеуге болады. Төменде бұл көрсетілетін болады.

Есептік жазбаның пайдаланушысына тиесілі нысандардың жиынтығы *сұлба* деп аталады. Жүйеге кіру құқығы жоқ пайдаланушыны жасауға

болады, алайда сұлбаны деректер қорының объектілерін жеке сақтау үшін қолданады.

Пайдаланушылар мен олардың аутентификациясының еределері SQL-операторлары *CREATE USER* және *ALTER USER* арқылы анықталады және өзгертіледі.

Пайдаланушылық akkaунттарды өзгерту. Пайдаланушылық есептік жазбаның параметрлерін өзгерту үшін тағы да Enterprise Manager-ді қолдануға болады немесе сәйкесінше SQL-команданы орындайды.

Үнсіз келісім және шектеу бойынша кестелік кеңістіктер. Әрбір пайдаланушыға үнсіз келісім бойынша кестелік кеңістік береді. Ол жерде пайдаланушы өзінің объектілерін сақтады. Кестелік кеңістікті үнсіз келісім бойынша құрған кезде немесе есептік жазбаны модификациялаған кезде көрсетуге болады:

```
CREATE USER ALL_ORACLE IDENTIFIED BY qwerty  
DEFAULT TABLESPACE USERS
```

немесе пайдаланушының деректерін өзгерту үшін

```
ALTER USER ALL_ORACLE DEFAULT TABLESPACE USERS;
```

Деректер қорында кестелік кеңістікті үнсіз келісім бойынша өзгерту үшін *ALTER DATABASE* сөйлемі қолданылады:

```
ALTER DATABASE DEFAULT TABLESPACE users
```

Пайдаланушы нысан құрған кезде, егер ол үшін кестелік кеңістікті үнсіз келісім бойынша анық көрсетілмеген болса, онда мәліметер базасы үшін үнсіз келісім бойынша құрылған кестелік кеңістіктер қолданылатын болады.

Деректер қорын құрғаннан кейін SYSTEM кестелік кеңістігін ешқашан қалдырмау ұсынылады. Не құру кезінде ауыстырыңыз, не деректер қорын құрғаннан кейін үнсіз келісім бойынша басқа кестелік кеңістік құрыңыз.

Кестелік кеңістікке шектеу. Пайдаланушы өзінің нысанын құрастырмастан бұрын, кеңістікті қолдану үшін рұқсат орнату керек. Кестелік кеңістік үшін квота пайдаланушы қолдана алатын максималды орын есебімен анықталады. Үнсіз келісім бойынша шектеу жоқ. Кестелік кеңістікті қолдануды *CREATE USER* конструкциясы арқылы шектеуге болады:

```
CREATE USER all_oracle IDENTIFIED BY "ALL_ORACLE123456"  
QUOTA 100M ON ALL_USERS
```

UNLIMITED кілттік сөзі пайдаланушы кестелік кеңістікте тек қана қолжетімді жерлермен шектеліп қалғанын көрсетеді.

Егер ДҚБЖ пайдаланушысының кестелік кеңістігіне квота тағайындамайтын болса, онда нысандарды құру, деректерді сақтау және т.б. мүмкіндіктері болмайды.

Пайдаланушыны жою оған тиесілі барлық нысандарды жояды, алайда оның рөлі мен мүмкіндіктері орнында қалады.

Пайдаланушының профилін белгілеу. Уақытша кестелік кеңістікке және үнсіз келісім бойынша кестелік кеңістікке қосымша ретінде, пайдаланушыға профиль тағайындалады [5]. Профильдің құрамында екі нәрсе туралы

ақпарат болады, біріншісі – ресурстарды қолдануға шектеу, және екіншісі – құпия сөз арқылы басқару ережелері.

Үнсіз келісім бойынша профиль дәл осылай – default деп аталады. Пайдаланушының профилін нақты белгілеу үшін CREATE USER немесе ALTER USER сөйлемдерін PROFILE кілттік сөзіне қосады да, профильдің атын көрсетеді. Мысалы:

```
CREATE USER all_oracle IDENTIFIED BY qwerty
DEFAULT TABLESPACE USERS
TEMPORARY TABLESPACE TEMP
PROFILE resource_profile;
```

немесе

```
ALTER USER all_oracle
PROFILE resource_profile;
```

Профильдер екілік функцияға ие, бұл парольдік саясатты іске асыру және ресурстарды анықтау. Парольдік саясат әрқашан орындалады, ресурстарды пайдалануды бақылау жүзеге асырылады, егер RESOURCE_LIMIT параметрінің мәні TRUE тең болса, ол үнсіз келісім бойынша FALSE тең болады. Профильдер автоматты түрде пайдаланылады, алайда үнсіз келісім бойынша барлық пайдаланушыларға белгіленетін профильдер, соның ішінде SYS, SYSTEM және т.б. пайдаланушылар үшін, – DEFAULT өте қарапайым.

Профильді өзгерту. Кез келген профильді сіз SQL-командасы арқылы ALTER PROFILE (сіз біратты мүмкіндіктерге ие болуыңыз керек) өзгерте аласыз. Профильді өзгерту кезінде ағымдағы сессия үшін шектеу өгермейтініне назар аудару керек. Өзгерістер өз күшіне енуі үшін жаңа сессияны күшіне енгізу керек. Команданың мысалы:

```
ALTER PROFILE clerk LIMIT
CPU_PER_CALL default
LOGICAL_READS_PER_SESSION 20000
```

Ресурстардың шектеулігін қосу үшін («заработал» профилі) орындаймыз:

```
ALTER SYSTEM SET RESOURCE_LIMIT=TRUE
немесе
ALTER SYSTEM SET RESOURCE_LIMIT=TRUE SCOPE = BOTH
```

Пайдаланушының мүмкіндіктері. DDL - GRANT (апарып беру) және REVOKE (кері қайтару) операторларын PL/SQL арқылы тікелей пайдалануға болмайды. Олар басқа басқа SQL операторлардың мүмкіндіктерін орындау үшін арналған. Мысалы, Oracle кестесінде кейбір операцияларды орындау үшін - INSERT немесе DELETE, GRANT операторларына мүмкіндік жасалуы керек. Екі әр алуан түрлі мүмкіндіктер болады: нысандық және жүйелік. Нысандық мүмкіндік (object privilege) нақты нысан үстінен белгілі бір операциялар орындауға мүмкіндік береді (мысалы, кесте үстінен). Дәл осы уақытта жүйелік мүмкіндік секілді (system privilege) нақты кластар

нысанынан операцияларды орындауға мүмкіндік береді. *DDL* мүмкін операциялардың барлық тәжірибесіне сәйкес келетін көптеген жүйелік мүмкіндіктер бар.

DDL тілінде негізгі операциялар үшеу - бұл *CREATE, ALTER, DROP*.

CREATE тобы:

Нақты сұлбада сәйкес нысандар құруға мүмкіндік береді:

CREATE PROCEDURE, CREATE SEQUENCE, CREATE TABLE, CREATE TRIGGER, CREATE VIEW, CREATE INDEX, CREATE SESSION, CREATE ROLE;

ALTER тобы:

ALTER DATABASE – БД-ның өзімен өзгертуге мүмкіндік береді.

ALTER USER – Пайдаланушы мен оның параметрлерін өзгертуге мүмкіндік береді (пароль, профиль, рөл және т.б.)

ALTER PROFILE – Профильді өзгертуге мүмкіндік береді.

ALTER TABLESPACE – Кестелік кеңістіктерді өзгертуге мүмкіндік береді.

DROP тобы кез келген сұлбада нысандарды жою, сондай-ақ кестелерді тазалайды.

DELETE ANY TABLE, DROP ANY PROCEDURE, DROP ANY SEQUENCE, DROP ANY TABLE, DROP ANY TRIGGER, DROP ANY VIEW, DROP ANY INDEX.

Мүмкіндіктер қалай өзгереді, немесе жойылатынын қарастырайық. Бұл *REVOKE* операторын қолдану үшін керек. Оның синтаксисі аз ғана айырмашылықтармен бірінші екі операторға ұқсас болады:

*REVOKE {privilege [, privilege...]}|ALL} ON object
FROM {user[, user...]}|role|PUBLIC} [CASCADE CONSTRAINTS];*

Мысалы, мүмкіндікті *SALESREPS* кестесінен *DUMMY* сұлбасына арнап таңдау үшін *RUSTEM* сұлбасына келесілерді енгізіңіз:

*REVOKE SELECT ON SALESREPS FROM DUMMY
/*

3 Зертханалық жұмыс № 3. ДҚБЖ ORACLE деректер қорларының объектілерін құру

Жұмыстың мақсаты: кестелік кеңістіктер мен деректер қорының объектілерін құруды үйрену.

3.1 Зертханалық жұмысқа тапсырмалар

1. MY_WORKER кестесін құрыңыз.

Name	Null?	Type
ID	NOT NULL	NUMBER(4)
LAST_NAME		VARCHAR2(25)
FIRST_NAME		VARCHAR2(25)
USERID		VARCHAR2(8)
SALARY		NUMBER(9,2)
DEPT_ID		NUMBER(3)

2. Бағаналардың атын анықтау үшін құрылым енгізіңіз.
3. MY_WORKER кестесінің *бірінші қатарына* келесі деректерді енгізу үшін INSERT құрыңыз. INSERT-тегі бағаналарды санамаңыз. *Барлық қатарларды енгізбеу керек.*

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	raptel	895
2	Danes	Betty	bdancs	860
3	Biri	Ben	bbiri	1100
4	Newman	Chad	cnewman	750
5	Ropeburn	Audrey	aropebur	1550

4. Алдыңғы тізімнен екінші жол үшін MY_WORKER кестесін толтырыңыз. Бағаналарды INSERT-те есептеңіз.
5. Кестеге толықтауыштарды тексеріңіз (select).
6. MY_WORKER кестесіне жолдарды сақтау үшін, файл сценариінде қою операторын қайта пайдалануға жазыңыз. Сақтаңыз script_03.sql.
7. Алдыңғы тізімнен кестедегі келесі екі жолды Сіз құрастырған сценариідегі қою операторын пайдаланы отырып, толтырыңыз.
8. Кестеге қосымшаны тексеріңіз.
9. Тұрақты деректерге қосымша жасаңыз. MY_WORKER кестесінде деректерді жаңартыңыз және жойыңыз.
10. MY_WORKER кестесіндегі кейбір деректерді жасыру керек. Ұсынысты WORKERS_VU деген атпен жасаңыз да, оны қызметкердің нөміріне, атына және WORKERS кестесіндегі бөлімнің нөміріне негіздеңіз.
11. Экранға WORKERS_VU ұсынысынан шыққан мазмұнды енгізіңіз.

Төменде орындау нәтижелері көрсетілген.

```

WORKER_ID  WORKER      SECTION_ID
-----

```

```

198      OConnell    50
199      Grant      50
200      Whalen     10
201      Hartstein  20
202      Fay       20

```

12. DEPT кестесінің біріншілік кілтiнiң бағанасы үшін пайдаланылуы мүмкiн тiзбек жасау керек. Тiзбек 200-ден басталуы керек және максималды мәні 1000 болуы керек. Тiзбектiлiктiң инкрементi 10. DEPT_ID_SEQ тiзбегiн атаңыз.

13. DEPT кестесіне екі жол қою үшін сценарий жазыңыз. script_10_08.sql сценарийін атаңыз. Екі бөлім қосыңыз: Education және Administration. Өзгерістерді тексеріңіз. Сценарийді орындаңыз.

Курстық жоба тапсырмасы бойынша деректер қорын құру, 3-4 кестедеге біріншілік кілттің автоматты генерациясының 2 тізбектілігін

қолдана отырып жүзеге асады. Зертханалық жұмыс мысалын пайдаланып, деректер базасы толтырылады.

Бақылау сұрақтары:

- 1) DESCRIBE операторы не үшін қолданылады?
- 2) Кесте дегеніміз не?
- 3) Кесте қандай элементтерден тұрады?
- 4) Кесте деректерінің типін зерттеңіз
- 5) 0 және NULL мәндерін қамтитын жолдар арасында қандай айырмашылық бар?
- 6) CREATE TABLE операторының көмегімен имя_таблицы AS SELECT кестеде көшірме жасалуы мүмкін бе?
- 7) DROP COLUMN командасынан бас тартып, бағананы кестеге қайтаруға бола ма?
- 8) Қандай SQL командалары арқылы ORACLE нысандары жасалады?
- 9) Қандай SQL командалары арқылы ORACLE нысандары жойылады?
- 10) Қандай SQL командалары арқылы ORACLE нысандары модификацияланады?
- 11) Қандай команда арқылы ORACLE бағанасын қосуға/жоюға болады?
- 12) Қандай команда арқылы жолын қосуға/жоюға болады?
- 13) ORACLE кестесінен деректер қандай SQL командалары арқылы таңдалады?
- 14) Қандай SQL командалары арқылы деректер қорының бастапқы күйіне шегініс жасалады?
- 15) Қандай SQL командалары арқылы транзакцияны жабу орындалады?
- 16) Қандай SQL командалары арқылы деректер қорында жазулар өзгерісі орындалады?

3.2 Зертханалық жұмысты орындауға арналған әдістемелік нұсқаулар

Жұмысты орындау реті.

1. Кестелермен жұмыс.

a) Кесте құрыңыз.

```
CREATE TABLE dept (deptno NUMBER(2), dname VARCHAR2(14),  
loc VARCHAR2(13), COMM VARCHAR2(20),);
```

b) Келесі мысалда DEPT кестесінің COMM бағанына қайта ат беріңіз:

```
ALTER TABLE dept RENAME COLUMN COMM TO COMMENTS;
```

c) Шектеуді өшіріңіз.

```
ALTER TABLE employees DISABLE CONSTRAINT emp_dt_fk;
```

d) DDL командасының көмегімен HR сұлбасында mySequence1 тізбегін құрыңыз. Бұл тізбек 100 нөмірінен басталуы керек және тізбек бойымен келіп, ең азы 10-ға дейін өсуі керек. Тізбектің басқа параметрлері үшін мәнді үнсіз келісім бойынша қалдырыңыз.

```
CREATE SEQUENCE mySequence1
```

*INCREMENT BY 10
START WITH 100*

2. Синонимдер.

а) DEPT_SUM_VU ұсынысы үшін әлдеқайда қысқа атау беріңіз.

```
CREATE SYNONYM d_sum FOR dept_sum_vu
```

б) Алиса пайдаланушысына тиесілі болатын DEPT кестесі үшін ортақ пайдалануға тиесілі синонимін DEPT құрыңыз:

```
CREATE PUBLIC SYNONYM dept FOR alice.departments
```

3. Ұсыныстар жасау.

50 бөлімшесінде жұмыс жасайтындар үшін ақпаратпен EMPVU80 ұсынысын жасау.

```
CREATE VIEW      salvu50  
AS  
SELECT employee_id IDNUMBER, last_name NAME,  
       salary*12 ANN_SALARY  
FROM employees WHERE department_id = 50
```

Кестелік кеңістік.

Oracle-дағы кестелік еңістік — бұл деректерді сақтаудың логикалық құрылымы [4]. Өңдеушілер кестелік кеңістіктерде сегменттер құрады. Кестелік кеңістік – абстракцияның жоғары деңгейі. Келесі қасиеттерге ие:

1) Oracle деректер базасы кестелік кеңістіктер деп аталатын әлдеқайда ұсақ облыстарға логикалық топтастырылуы мүмкін.

2) Кестелік кеңістіктер бір ғана деректер қорына жатуы мүмкін.

3) Әрбір кестелік кеңістіктер бір немесе бірнеше деректер файлынан тұрады.

4) Кестелік кеңістіктерде бір немесе бірнеше деректер сегменті болуы мүмкін.

5) Кестелік кеңістіктер белсенді ережеге (online) ауыстырылуы мүмкін, ол жерде деректер пайдаланушылар үшін қолжетімді, немесе белсенді емес (автономды, offline) ереже, ол жерде файлдар жабық болады және деректер пайдаланушылар үшін қолжетімсіз.

6) Кестелік кеңістіктерді «тек қана оқу» (Read only) режиміне немесе жазу (read write) режиміне ауыстыруға болады.

Oracle ДҚ құру үдерісінде автоматты түрде, пайдаланушылар үшін арнайы SYSTEM кестелік кеңістігі құрылады. Үлкен емес деректер базасы үшін де осы кестелік кеңістік жетіп жатыр, алайда бәрібір қосымша кестелік кеңістіктерді мәліметті пайдаланушылар үшін құрған жөн. SYSTEM кестелік кеңістігінде деректер сөздігі сақталады. Деректер сөздігінде кестелер, индекстер, кластерлер және т.б. туралы ақпараттар болады.

Деректер қорын құрғаннан кейін SYSTEM кестелік кеңістігін ешқашан үнсіз келісім бойынша қалдырмауды ұмынады. Не деректер қорын құру кезінде өзгертіңіз, не мәліметер қорын құрғаннан кейін басқа кестелік кеңістікті үнсіз келісім бойынша көрсетіңіз.

Кестелік кеңістікті құрған кезде файлдың қандай типі, үлкен немесе кіші екендігі, экстендтерді пайдаланудың қандай типі, локальді ма, әлде деректер сөздігі ме, және де кеңістіктік сегменттермен басқару – автоматты немесе қолмен жүргізілетінін таңдап алу керек. Кестелік кеңістіктерді құру, жою және өзгерту үшін, пайдаланушы келесі мүмкіндіктерге ие болуы керек - *CREATE TABLESPACE, DROP TABLESPACE, ALTER TABLESPACE*.

Администратор басшылығынан команданың толық конструкциясын қарастырайық:

CREATE TABLESPACE -- (кестелік кеңістік)

DATAFILE (файл_спецификациясы) (автокеңейтілуі)-*

Спецификациясы

DEFAULT [COMPRESS|NOCOMPRESS] STORAGE storage_clause

MINIMUM EXTENT int {K|M}

BLOCKSIZE int K

LOGGING | NOLOGGING

FORCE LOGGING

ONLINE | OFFLINE

PERMANENT | TEMPORARY

EXTENT MANAGEMENT {DICTIONARY |

LOCAL {AUTOALLOCATE | UNIFORM [SIZE int K | M]}}

SEGMENT SPACE MANAGEMENT {MANUAL | AUTO}

Oracle 10g-дан бастап, үлкен файлдары бар кестелік кеңістіктер пайда бола бастады (bigfile). Мұндай кестелік кеңістіктер бір деректер файлында немесе уақытша файлда салынады. Құрамында 2^{32} деректер блогына дейін болуы мүмкін. Сондай-ақ кестелік кеңістіктер 8К деректер блогын пайдалана отырып, 32ТВ-қа дейінгі көлемді алуы мүмкін.

Мұндай кестелік кеңістіктер өте үлкен деректер базасы үшін пайдаланылады. Мұндай кестелік кеңістікті құру үшін BIGFILE кілттік сөзімен CREATE сөйлемі орындалады.

Немесе ALTER USER сөйлемін пайдалана отырып, бар пайдаланушының квотасын өзгерту.

Ішкі сұрауларды пайдалану арқылы кесте құру. Кестелер құрудың екінші әдісі – бұл AS *subquery* сөйлемін кестелер үшін де, сондай-ақ оны ішкі сұрауларға қайтатындай етіп (*subquery*) жолдармен толтыру үшін де пайдалану.

Көрсетілімдер:

1) Кесте берілген баған атауларымен жасалады, және оған SELECT командасы арқылы қайтарылатын жолдар қойылады.

2) Бағанның анықтамасы тек бағанның атын және үнсіз келісім бойынша оның мәнін береді.

3) Егер жаңа кесте үшін бағананың сипаттамасы берілген болса, онда олардың саны ішкі сұраудың (SELECT) таңдау тізімінің санына сәйкес келуі керек.

4) Егер бағаналардың сипаттамасы берілмеген болса, онда олардың аты ішкі сұраудағы бағаналардың атауы сияқты болады.

```
CREATE TABLE      dept80
AS
  SELECT  employee_id, last_name,
          salary*12 ANNSAL, hire_date
  FROM    employees
  WHERE   department_id = 80;
```

Кестелерді редактрлеу.

Кестелерді редактрлеу үшін *SQL* – ALTER TABLE конструкциясы қолданылады.

```
ALTER TABLE table ADD|MODIFY|DROP (column datatype [DEFAULT
expr] [, column datatype]...)
```

Белгілі бір шарттарды орындау үшін әр түрлі сөйлемдерді пайдалануға болады *ON DELETE CASCADE| ON DELETE SET NULL*.

```
ALTER TABLE employ ADD CONSTRAINT emp_mgr_fk
FOREIGN KEY(manager_id)
REFERENCES employ(employee_id) ON DELETE CASCADE
```

Пайдаланушы кесені редактрлей алуы үшін, ол оның сұлбасында болуы керек, ал пайдаланушыда нысандық мүмкіндік ALTER TABLE болуы керек, кестелерді редактрлеу үшін басқа пайдаланушыларға ALTER ANY TABLE жүйелік мүмкіндігі керек болады.

Кестеге бағанадар қосу.

Бар кестеге баған қосу үшін ALTER TABLE ... ADD конструкциясы пайдаланылады. Келесі мысалда бар кестені өзгертеміз ALL_ORACLE.ADMIN_LOG, және COMMENTS бағанын қосамыз:

```
ALTER TABLE ALL_ORACLE.ADMIN_LOG ADD (COMM VARCHAR2 (100))
```

Кестенің бағандарын жою.

Енді керек қылмайтын бағандарды жоюға болады. Бұл деректер қорындағы кеңістікті босатыдың жақсы амалы. Кестеден барлық бағандарды бірден өшіріп тастауға болмайды, сондай-ақ кестеден бағанды өшіру SYS пайдаланушысына тиесілі.

Жою үшін ALTER TABLE ... DROP COLUMN конструкциясы пайдаланылады. Нәтижесінде бағанның дискрипторы және бағанмен байлансты деректер жойылады. Бір конструкцияда бірден бірнеше бағанды өшіруге болады. Мысалдар мынаны көрсетті:

```
ALTER TABLE ALL_ORACLE.ADMIN_LOG DROP COLUMN COMMENTS;
```

Екі бағанды бірден өшіру DATE_TIME және EVENT

```
ALTER TABLE ALL_ORACLE.ADMIN_LOG DROP (DATE_TIME , EVENT)
```

Кестенің бағандарының атауын алмастыру.

Oracle кестеде бар бағанның атауын ауыстыруға мүмкіндік береді. Атауын қайта ауыстыру үшін ALTER TABLE-дегі RENAME COLUMN сөйлемі қолданылады. Бағанның жаңа атауы бұрыннан бар бағанмен бірдей болып қалмауы керек. RENAME COLUMN сөйлемімен бірге басқа сөйлемдерді көрсетуге болмайды.

```
ALTER TABLE marketing RENAME CONSTRAINT mktg_pk TO new_mktg_pk
```

Шектеулерді қосу. Шектеулер кесте деңгейінде ережелердің орындалғандығын қамтамасыз етеді. Шектеу басқа кестелерде бағыныңқы деректер бар болған жағдайда кестенің жойылуының алдын алады.

Oracle-да келесі шектеулерге жол беріледі:

```
NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY, CHECK.
```

DISABLE/ENABLE операторында ALTER TABLE командасын пайдалана отырып, шектеуді жоймай-ақ өшіруге/қосуға болады.

```
ALTER TABLE table DISABLE CONSTRAINT constraint [CASCADE];
```

Табыстау. Табыстаудың мүмкіндіктері:

1) Деректер қорына қолжетімділікті шектейді, яғни табыстау пайдаланушыға деректер қорының тек оның қолжетімді бөлігін ғана көрсетуіне болады.

2) Пайдаланушыға қарапайым сұраулардың көмегімен күрделі сұраулардың нәтижелерін алуға мүмкіндік береді. Мысалы, пайдаланушылар, бірнеше кестеден бірден ақпарат сұрауына болады

3) Жоспарларнбаған сұраулар жіберетін пайдаланушылар мен қолданбалы бағдарламалар үшін деректердің тәуелсіздігін қамтамасыз етеді. Бір табыстау бірнеше кестеден бір таңдамалыны пайдалануға мүмкіндік береді.

4) Пайдаланушылар тобына нақты берілген критерилер бойынша деректерге қолжетімділік жасайды.

Табыстаулардың қарапайым және күрделі түрлері арасында негізгі айырмашылық DML операцияларымен байланысты (көшіріп қою, жаңарту және жою).

Қарапайым табысау (simple view) – бұл табыстау:

1) Мәліметтерді тек бір кестеден таңдайды.

2) Құрамында функциялар, DISTINCT сөйлемі және деректер тобы болмайды.

3) Табыстау арқылы DML операциясын орындауға мүмкіндік береді.

Күрделі табысау (complex view) – бұл табысау:

1) Мәліметтерді бірнеше кестелерден таңдайды.

2) Қрамында функциялар, деректер тобы, DISTINCT сөйлемі бар.

3) Табыстау арқылы DML операциясын орындауға әрқашан да мүмкіндік бере бермейді.

Табыстау CREATE VIEW сөйлемі арқылы ішкі сұрауларды қосу арқылы жасалады:

```
CREATE [OR REPLACE] [FORCE|NOFORCE] VIEW представление
```

```
[(псевдоним[, псевдоним]...)]
```

```
AS подзапрос
```

```
[WITH CHECK OPTION [CONSTRAINT ограничение]]
```

```
[WITH READ ONLY [CONSTRAINT ограничение]]
```


Бағаналардың аттарын ішкі сұраудағы бағанаға бүркеншек аттар қосу арқылы (алиастар) басқаруға болады.

Бағандардың аттарын басқарудың тағы бір әдісі – бұл олардағы бүркеншек аттарын CREATE VIEW сөйлемінде табыстаудың атынан кейін көрсету.

```
CREATE VIEW salvu50 (ID_NUMBER, NAME, ANN_SALARY)
AS
SELECT employee_id, last_name, salary*12
FROM employees
WHERE department_id = 50;
```

Бағаналардың аттарын ішкі сұрауардың бағаналарына бүркеншек ат қосу жолымен басқаруға болады.

Бағаналардың аттарын басқарудың тағы бір жолы – бұл олардың бүркеншек аттарын CREATE VIEW сөйлемінде табыстаудың атынан кейін көрсету.

```
CREATE VIEW salvu50 (ID_NUMBER, NAME, ANN_SALARY)
AS
SELECT employee_id, last_name, salary*12
FROM employees
WHERE department_id = 50;
```

Табыстаумен DML операциясы арқылы орындау ережесі.

1. DML операцияларын қарапайым бабыстау арқылы орындауға болады.

2. Жолды жоюға болмайды, егер табыстаудың құрамында мыналар болса:

- 1) Топтық функциялар.
- 2) GROUP BY сөйлемі.
- 3) DISTINCT кілттік сөзі.
- 4) ROWNUM жалған бағанға сілтеме.

Құрамында мыналар бар табыстауларға деректерді қосуға болмайды:

- 1) Топтық функциялар.
- 2) GROUP BY сөйлемі.
- 3) DISTINCT кілттік сөзі.
- 4) ROWNUM жалған бағанға сілтеме.
- 5) Көрсеткіштердің көмегімен анықталатын бағандар.

Базалық кестелерде анықталатын бағандар NOT NULL шектеулері арқылы табыстауға деректер қосуға болмайды.

WITH CHECK OPTION сөйлемін пайдалану. DML операциялардың нәтижелері табыстаудың домен шегінде қалуын бақылау керек.

Тізбектіліктер.

Кестедегі жолдар қатарын автоматты түрде генерация жасау үшін сандар генераторын пайдалануға болады. Тізбектілік (SEQUENCE) – бұл пайдаланушы құрастыратын және бірнеше пайдаланушы тарапынан

ұжымдық түрде қолданыла алатын деректер қорының нысаны. Біріншілік кілттің мәнін генерациялау үшін жиі қолданылады.

Тізбектіліктің саны кестеге тәуелсіз түрде сақталады және генерацияланады. Сәйкесінше, бір тізбектілік бір уақытта бірнеше кесет үшін қолданыла алады.

Бұл өте ыңғайлы болады және жұмысшыдан біріншілік кілттің мәнін құрастыру үшін мән беру алгоритмін жүзеге асыруға мүмкіндік береді. Әрбір жаңа мән тізбекте құру кезіндегі қағидалардың негізінде жасалады. Құрудың ортақ синтаксисі:

```
CREATE SEQUENCE [schema].sequence_name
[INCREMENT BY number]
[START WITH number]
[MAXVALUE number | NOMAXVALUE]
[MINVALUE number | NOMINVALUE]
[CYCLE | NOCYCLE]
[CACHE number | NOCACHE]
[ORDER | NOORDER]
```

Тізбектілік мәндерін алу. Тізбектілік мәндерін алуды үйренеміз. Ең қарапайым мысал осы сұрауды орныдау болып табылады:

```
SELECT SEQ.NEXTVAL FROM DUAL
```

Оны деректерді кестеге салу кезінде біріншілік кілттің мәнін генерация жасау үшін пайдалануға тырысып көрейік:

```
INSERT INTO ALL_ORACLE_USERS (ID, NICK_NAME, REG_DATE)
VALUES (SEQ.NEXTVAL, 'USER_TEST', '01/01/2010')
```

Тізбектілік параметрлерін өзгерту.

Тізбектілік атрибуттарын өзгерту үшін ALTER SEQUENCE конструкциясы қолданылады. INCREMENT BY, MAXVALUE, MINVALUE, CYCLE и CACHE атрибуттарын өзгертуге болады, алайда START WITH өзгертілмейді. Мысалы:

```
ALTER SEQUENCE SEQ INCREMENT BY 2;
```

Егер START WITH атрибутын өзгерту керек болса, немесе ағымдағы тізбектілік мәнін сұрау керек болса, онда оны жою, жаңа атрибуттармен қайта құру, және де оған барлық керекті мүмкіндіктерді туғызу керек.

Тізбектілікті жою.

Жою үшін DROP SEQUENCE конструкциясы пайдаланылады. Мысалы:

```
DROP SEQUENCE SEQ;
```

Синонимдер базалық нысандардың атауын өзгерте алады және орнын алмасытар алады. Мұндайда синонимдер ғана қайта анықталады, ал қосымшаларда ешқандай өзгерістер болмайды.

Синонимдердің екі түрі бар:

1) Жеке (PRIVATE) – синонимдер нақты пайдаланушының сұлбасында көрінеді және тек пайдаланушының өзі үшін ғана, сонымен қатар ол өзі сәйкес рұқсаттылықтарды берген адамға ғана қолжетімді.

2) Ортақ (PUBLIC) – бұл синонимдердің иелері – арнайы пайдаланушылар тобы – PUBLIC, соның нәтижесінде бұл синонимдер деректер қорының барлық пайдаланушылары үшін қолжетімді.

Синонимдерді жасау.

Жеке синонимдерді жасау үшін CREATE SYNONYM мүмкіндігіне ие болу керек, басқа пайдаланушының сұлбасында синоним құру құқығын алу үшін CREATE ANY SYNONYM мүмкіндігі керек. Ортақ синоним құру үшін CREATE PUBLIC SYNONYM мүмкіндігі керек.

Синонимдерді құру үшін CREATE SYNONYM немесе CREATE PUBLIC SYNONYM конструкциялары пайдаланылады, ол құрылып отырған синонимнің түріне де байланысты. Нақты нысан болған кезде мүмкіндікке қолжетеркізу міндетті емес.

```
CREATE SYNONYM [имя_синонима] FOR [имя_объекта]
CREATE PUBLIC SYNONYM [имя_синонима] FOR [имя_объекта]
```

Синонимдерді жою.

Пайдаланушы өз сұлбасында синонимнің кез келген бөлігін жою алады. Басқа пайдаланушының сұлбасында жеке синонимді жою үшін DROP ANY SYNONYM мүмкіндігі керек, ал ортақ синонимді жою үшін – DROP PUBLIC SYNONYM мүмкіндігі керек.

Жою үшін – DROP SYNONYM операторы қолданылады. Ортақ синонимді жою үшін - DROP PUBLIC SYNONYM. Мысалы, ертеректе жаалған синонимдері жойық:

```
DROP SYNONYM SYN_ALL_ORACLE_USERS;
DROP PUBLIC SYNONYM SYN_ALL_ORACLE_USERS
```

Функцияларға негізделген индекстер.

UPPER (column_name) немесе LOWER (column_name) кілттік сөздері арқылы жарияланған Function-based индекстері іздестіру регистрына тұрақсыздықты қолдануға мүмкіндік береді. Мысалы, келесі индекс:

```
CREATE INDEX upper_last_name_idx
ON employees (UPPER(last_name))
```

Сұрау үдерісін жеңілдету үшін:

```
SELECT * FROM employees WHERE UPPER(last_name) = 'KING'
```

Oracle сервері кепілдік бере отырып, индексті пайдаланылуы үшін (толық сканерлеудің орнына кестелер), функцияның мәні келесі сұрауларда анықталмаған болып табылады. Мысалы, келесі доменда индексті пайдалануға кепілдік береді, алайда WHERE сипатынсыз Oracle сервері кестенің толық маскасын орындауға мүмкіндік береді:

```
SELECT * FROM employees WHERE UPPER (last_name) IS NOT NULL
```

```
ORDER BY UPPER (last_name)
```

Oracle сервері DESC арқылы *function-based* индекстері ретінде белгіленген колонналар арқылы өңдейді. DESC арқылы белгіленген колонналар ағымдағы емес ретпен саналады.

Индексті жою үшін DROP INDEX командасы пайдаланылады

```
DROP INDEX upper_last_name_idx
```

SQL тілінде деректерді модификациялау құралы. Деректерді модификациялау болуы мүмкін үш операцияны бөледі: қою (вставка), өзгерту және жою. Бұл бөлімде осы операциялар SQL тілінің құралдары арқылы қалай жасалатыны көрсетіледі.

Мәнді кестеге қою.

Мәнді кестеге қарапайым қою. Мәнді кестеге қою үшін SQL-оператор INSERT қызмет етеді. Бұл операторда екі форма бар, ол кестенің барлық бағаналарына мән тиесілі ме, соған байланысты. Егер барлық бағандардың мәндері көрсетілсе, онда оператор келесі түрге ие болады:

```
INSERT INTO hr.employees VALUES(900, 'Ivan', 'Ivanov',  
Upper(substr('Ivan',1,1) || 'Ivanov'), '555-55-55',sysdate,  
'SH_CLERK', null, null,null, null);
```

Егер, қандай да бір бағана үшін деректер жоқ болса, онда жіберілетін мәндер үшін NULL кілттік сөзін пайдалануға болады:

Кеңейтілген оператор INSERT пайдалануға болады.

```
INSERT [conditional_insert_clause]  
[insert_into_clause values_clause] (subquery)  
[ALL|FIRST] [WHEN condition THEN] [insert_into_clause  
values_clause] [ELSE] [insert_into_clause values_clause]
```

Бірнеше жолды бір кестеге буақыттық шартты қоюды орындау *INSERT ALL* (*SELECT* операторының командасынада қатысу міндетті) командасының көмегімен жүруі мүмкін

```
CREATE table tab1 (f1 number, f2 varchar2(255));
```

```
Table created.
```

```
INSERT ALL
```

```
INTO tab1 (f1,f2) VALUES (1,'One')
```

```
INTO tab1 (f1,f2) VALUES (2,'Two')
```

```
INTO tab1 (f1,f2) VALUES (3,'Three')
```

```
SELECT * FROM DUAL;
```

Егер басқа кесетеден үлкен көлемді деректерді көшіру талап етілсе, онда олардың мәндерін SELECT операторының көмегімен алуға болады.

```
INSERT INTO sales_reps(id, name, salary, commission_pct)  
SELECT employee_id, last_name, salary, commission_pct  
FROM employees WHERE job_id LIKE '%REP%';
```

Мұндай жағдайда VALUES кілттік сөзі SELECT операторының құрамына кіретініне назар аударыңыз.

Деректердің өзгеруі.

Бар деректердің мәндері SQL-операторының UPDATE көмегімен өзгертілуі мүмкін. Назарға алу керек, бұл мықты команда және оның абайлап пайдалану керек.

Келесі мысалды қарастырайық.

Қызметкердің нөмірі мен жаңа жалақысы туралы ақпарат сұрайтын Oracle SQL командасын жазайық, және де алынған ақпараттың негізінде кестеге өзгеріс енгізейік hr.employees (бұл қызметкер үшін жалақыны өзгерткен). Қызметкерің нөмірі employee_id бағанасында, ал жалақысы — salary бағанасында көрсетілген.

Ескерту: жіберіледі, алдымен жаңа жалақы туралы сұрау, ал содан соң – бағанның аты туралы сұрау.

Осы өзгерісті енгізе отырып, сіз қате жібердіңіз және WHERE сөйлемі туралы ұмытып кеттіңіз дейік. Соңында, егер salary бағанының шектеуі бұзылса (мысалы, уникальдылығы бойынша шектеу), онда барлық EMPLOYEES кестесіндегі salary бағанында ионитордан енгізілген бір мән болады.

Кестені сұрауды пайдалану арқылы жаңартуға болады.

Деректерді жою.

Жолдарды жою үшін SQL-оператор DELETE қызмет етеді. Оған K UPDATE операторына жататын мүмкіндіктре жатады. Ол пайдаланан кезде алдамшы жеңіл, және оны абайсыз қолдану ең күтілмеген нәтижелерге алып келуі мүмкін.

Тексеру үшін нөмірі 900 болатын қызметкердің жазбасын өшіріңіз (ол алдыңғы зертханалық жұмыстардың бірінде құрылған болатын). Жою орныдалғаннан кейін COMMIT командасын орындаңыз.

```
DELETE * FROM hr.employees WHERE employee_id=&Номер_сотрудника
```

UPDATE операторымен болған жағдай секілді, егер сіз WHERE сөйлемін ұмытып кетсеңіз, оның нәтижесі апатты болуы мүмкін. Мысалы, келесі оператор hr.employees қатынасындағы барлық жолдарды өшіріп тастайды:

```
DELETE * FROM hr.employees
```

4 Зертханалық жұмыс № 4. SQL ДҚБЖ ORACLE арнайы операторлары мен функциялары

Жұмыстың мақсаты: SQL ДҚБЖ ORACLE операторлары мен функцияларының ерекшеліктерін зерттеу.

4.1 Зертханалық жұмысқа тапсырмалар

Зертханалық жұмыста келтірілген барлық тапсырмаларды орындау.

1. 1998 жылдың 20 ақпаны мен 1998 жылдың 1 мамыры аралығындағы кезеңде жалданған барлық қызметкерлердің тегін, қызметін және жұмыс жасауды бастаған күнін енгізіңіз. Жалдану уақытының өсу ретімен мәліметерді сұрыптаңыз.

2. Жалақысы \$2,500, \$3,500 немесе \$7000 тең емес сату бөлімінің өкілдері (sales representative) немесе биржалық маклер (stock clerk) болып жұмыс істейтін қызметкерлердің тегін, қызметі мен жалақысын сұратыңыз.

3. Қызметкерлердің тегін енгізіңіз (тегінің бірінші әрібі бас әріппен жазылуы керек, ал қалғандары – жазу әріптері) және тегі J, A немесе M әріптерінен басталатын барлық тектердің ұзындығын жазыңыз. Сәйкес тақырыптарды бағандармен сәйкестендіріңіз. Тектер бойынша нәтижелерді сұрыптаңыз.

4. Әрбір қызметкер үшін текті енгізіңіз және жалдау күнінен бастап, бүгінгі күнге дейінгі уақытты бүтінге жақын қылып дөңгелкетеңіз. Бағанды MONTHS_WORKED деп атаңыз. Нәтижелерді жұмыс жасалған айлардың саны бойынша сұрыптаңыз. Айлардың санын бүтін санға дейін дөңгелектеңіз.

5. Әрбір қызметкер бойынша мына түрдегі есепті алыңыз: <employee last name> earns <salary> monthly but wants <3 times salary> (<тегі> табады <жалақы> айына, бірақ алып қалады <үш еселік жалақы>). Бағанды Dream Salaries деп атаңыз.

6. Барлық қызметкерлердің тегін және жалақысын енгізу үшін сұрау жіберіңіз. Шығу бағанын SALARY деп атаңыз. Бағанның ұзындығы SALARY – 15 символдан тұрады, символдарды \$ сол жағынан толтырады.

7. Әрбір қызметкердің тегін, жалдану уақытын және жалақысын қарау күнін енгізіңіз, ол алты айлық қызметтен кейін бірінші дүйсенбіге сай келеді. Бағананы REVIEW деп атаңыз. Күндер (даталар) олардың шығуы келесіге ұқсас болатындай етіп форматталуы керек: “Monday, the Thirty-First of July, 2000”

8. Әрбір қызметкер бойынша оның тегін, ол жұмысқа алынған жалдану күнін және апта күнін енгізіңіз. Соңғы бағананы DAY деп атаңыз. Алынған нәтижелерді апта күндері бойынша, дүйсенбіден бастап орналастырыңыз.

9. Әрбір тектің шығуы бойынша және әрбір қызметкерлер үшін комиссиялық сумманы шығарыңыз. Егер қызметкер комиссиялық сумма алмаса, онда оны «No Commission» бағанасында көрсетіңіз. Бағананы COMM деп атаңыз.

10. Жұлдызшалармен белгіленген қызметкерлердің тегі мен олардың жалақыларын енгізу үшін сұрау жіберіңіз. Әрбір жұлдызша 1000 долларды білдіреді. Шығу деректерін жалақылардың кему реті бойынша іріктеңіз. Бағананы EMPLOYEE_AND_THEIR_SALARIES («Қызметкерлер мен олардың жалақылары») деп атаңыз.

11. DECODE функциясын пайдалана отырып, келесі деректер бойынша

JOB_ID бағанының негізінде барлық қызметкерлердің категориясын енгізу үшін сұрау жіберіңіз:

<u>JOB</u>	<u>GRADE</u>
AD_PRES	A
ST_MAN	B
IT_PROG	C
SA_REP	D
ST_CLERK	E
None of the above	0

Бақылау сұрақтары:

- 1) SQL*Plus қосымшасы мүмкіндік береді:
 - a) Oracle деректер қорынан деректерді бейнелеуге.
 - b) Деректерді есептер түрінде безендіруге.
 - c) Деректер қорының функцияларына администраторлық жасауға.
 - d) Жоғарыда айтылғандардың барлығы.
- 2) SQL буферімен жұмыс жасаған кезде I командасы INSERT командасының қысқартылған нұсқасы ретінде пайдаланылады.
Ия/жоқ
- 3) Үнсіз келісім бойынша ORDER BY фразасында өсу ретімен іріктеу жүргізіледі.
Ия/жоқ
- 4) Келесі LIKE командалардың қайсысы SOTKA атындағы жолдарды қайтарады?
 - a) WHERE ename LIKE '_SOT%'
 - b) WHERE ename LIKE 'S_T%'
 - c) WHERE ename LIKE '%TKA_'
 - d) WHERE ename LIKE '%TOK%'
- 5) Келесі UPDATE операторларының қайсысы жіберіледі?
 - a) UPDATE emp SET sal = sal * 1.05
 - b) UPDATE emp WHERE sal = 20000
 - c) UPDATE emp SET ename = 1234 WHERE deptno = 10
 - d) UPDATE emp WHERE deptno = 10 SET ename = 'SOTKA'

4.2 Зертханалық жұмысты орындауға арналған әдістемелік нұсқаулар

Жұмыстың орындалу реті.

1) hr сұлбасындағы employees кестесінде әрбір қызметкердің қызметтік мансабы үшін жалақының максималды, минималды және орташа шамасы туралы ақпарат бере алатын сұаныс жазыңыз. Жалақы туралы ақпарат salary бағанында орналасады, ал қызметтік мансап туралы ақпарат — job_id кестесінде.

Сәйкес сұраудың коды былай көрінуі мүмкін:

```

SELECT job_id AS "Должность", max(salary) AS "Максимальная зарплата",
MIN(salary) As "Минимальная зарплата", TO_CHAR(avg(salary), '99999999.99') AS
"Средняя зарплата"
FROM hr.employees
GROUP BY job_id;

```

2) hr сұлбасындағы employees кестесінде әрбір қызметкердің қызметтік мансабы үшін жалақының максималды, минималды және орташа шамасы тураы ақпарат бере алатын сұрау жазыңыз. Бұл кезде берілген ақпарат:

- алдымен мансабы бойынша топтастырылуы керек (employees кестесінің job_id бағаны);

- содан соң бөлімшелер бойынша топтастырылу керек (departments кестесінің department_name бағаны);

- жалақысының орташа мөлшері 7000-нан артық немесе соған тең мансаптар үшін топтастырулар жүргізіледі;

- ақпарат алдымен бөлімшелер бойынса, содан кейін мансаптар бойынша іріктелуі керек.

Сәйкес сұраудың коды былай көрінуі мүмкін:

```

SELECT departments.department_name AS "Отдел", employees.job_id AS
"Должность", MAX(employees.salary) AS "Максимальная зарплата",
MIN(employees.salary) AS "Минимальная зарплата",
TO_CHAR(avg(employees.salary), '99999999.99') AS "Средняя зарплата"
FROM hr.employees, hr.departments
WHERE hr.employees.department_id = hr.departments.DEPARTMENT_ID
GROUP BY hr.departments.department_name, hr.employees.job_id
HAVING AVG(employees.salary) >= 7000
ORDER BY hr.departments.department_name, hr.employees.job_id

```

3) Келесі сұрауды қолдана отырып, қызметкерлер туралы деректерді енгізіңіз

```

SELECT employee_id, CONCAT(first_name, last_name) NAME, job_id, LENGTH
(last_name), INSTR(last_name, 'a') "Contains 'a'?"
FROM employees WHERE SUBSTR(job_id, 4) = 'REP';

```

4) Бөлімшелердің атауын сол жақтан «_» символы арқаылы, енгізілетін атау қосымша символдармен бірге 20 символдан аспайтындай етіп қою керек.

```

SELECT LPAD(department_name, 20, '_')
FROM departments;

```

5) Символдар регистрін түрлендірудің үш функциясын пайдаланыңыз

a)

```

SELECT 'The job id for ' || UPPER(last_name) || ' is ' || LOWER(job_id) AS
"EMPLOYEE DETAILS"

```

```

FROM employees;

```

b)

```

SELECT employee_id, UPPER(last_name), department_id
FROM employees

```



```
WHERE INITCAP(last_name) = 'Higgins';
```

б) Даталармен жұмыс.

а) Аптаның күнін анықтау үшін TO_CHAR функциясын мына түрде пайдаланыңыз:

```
SELECT TO_CHAR(to_date('26.12.2012','dd.mm.yyyy'), 'd')  
FROM DUAL;
```

Нәтижесінде апта күндерінің реттік нөмірін аламыз: 1-ден 7-ге дейін.

Қандай күнді бірінші деп санайтыны NLS_TERRITORY параметріне байланысты болады.

б) «Осы аптаның бірінші күні қайсы күн» деген сұраққа мысан сұрауды толтырыңыз:

```
SELECT TRUNC(to_date('21.11.2011','dd.mm.yyyy'),'D') FROM DUAL;
```

с) Ағымдағы жылды енгізіңіз

```
SELECT EXTRACT ( YEAR FROM SYSTIMESTAMP ) FROM dual;
```

д) Есептеңіз

```
SELECT MONTHS_BETWEEN (DATE '2013-03-01', DATE '2014-02-28')  
FROM dual;
```

е) Уақыт моментін беру форматын ДҚ, ДҚБЖ және жеке сеанстар үшін орындауға болады. DATE типіне жақын сұрауларды орындаңыз:

```
SQL> SELECT value FROM nls_session_parameters  
2> WHERE parameter = 'NLS_DATE_FORMAT';  
VALUE
```

```
-----  
DD-MON-RR
```

```
SQL> SELECT SYSDATE FROM dual;  
SYSDATE
```

```
-----  
14-SEP-09
```

```
SQL> ALTER SESSION SET NLS_DATE_FORMAT = 'YYYY-MM-DD  
HH24:MI:SS';  
Session altered.
```

7) Деректерді қайта түрлендіру.

а) выдайте дату по формату

```
SELECT employee_id, TO_CHAR(hire_date, 'MM/YY') Month_Hired  
FROM employees  
WHERE last_name = 'Higgins';
```

Егер сіз апта күндері бойынша қайта сұрыптағыңыз келсе, онда ORDER BY-ға назар аударамыз, мысалы:

```
ORDER BY TO_CHAR(hire_date-1, 'D');
```

б) Формат бойынша қызметкерлердің жалақысын үлестіріңіз

```
SELECT employee_id, TO_CHAR(salary, '$99,999.00') SALARY  
FROM employees  
WHERE last_name = 'Smit';
```

с) Форматтардың алуан түрін пайдаланыңыз.

```
SELECT employee_ID, TO_CHAR(hire_date, 'DD, MM, Mon, Month, YYYY, Year,  
Day, DY, Q')
```

```
AS "All"  
FROM Employees;
```

Мысалдар:

```
SELECT ename, comm, NVL ( comm, 0 ) FROM emp;  
SELECT comm, sal, COALESCE ( comm, sal ) FROM emp;
```

8) Мәнді шартты қою функциялары.

a) COMMISSION_PCT кколоннасын тексеріңіз. Егер мән табылса, онда екінші көрсеткіш SAL+COMM қайтып оралады. Егер COMMISSION_PCT колонкасының құрамында NULL мәні бар болса, онда үшінші көрсеткіш SAL қайтып оралады.

```
SELECT last_name, salary, commission_pct,  
       NVL2(commission_pct, 'SAL+COMM', 'SAL') income  
FROM employees  
WHERE department_id IN (50, 80);
```

b) Екі көрсеткішті пайдаланыңыз

```
SELECT first_name, LENGTH(first_name) "expr1",  
       last_name, LENGTH(last_name) "expr2",  
       NULLIF(LENGTH(first_name), LENGTH(last_name)) result  
FROM employees;
```

c) Сұрауды орындаңыз – егер COMM комиссиялық шығындардың мәні null тең болса, ол көрінеді. Егер COMMISSION_PCT мәні null тең болса, онда SALARY жалақысы көрсетіледі. Егер COMMISSION_PCT және SALARY мәндері null тең болса, онда 10 мәні беріледі:

```
SELECT last_name, employee_id,  
       COALESCE(TO_CHAR(commission_pct), TO_CHAR(manager_id) ,  
       'No commission and no manager') comm  
FROM employees;
```

9) Шартты сұраулар жасаңыз.

a) Коэффициенттерге тәуелді түрде жалақыны есептеңіз

```
SELECT last_name, job_id, salary,  
       CASE job_id  
         WHEN 'IT_PROG' THEN 1.10*salary  
         WHEN 'ST_CLERK' THEN 1.15*salary  
         WHEN 'SA_REP' THEN 1.20*salary  
         ELSE salary  
       END "REVISED_SALARY"  
FROM employees;
```

b) Жолдар (қызметкерлер) мен бағаналардың (бөлімшенің атауы) қиылысқан жеріне осы бөлімшенің қызметкерінің жалақысын қою керек (тек 10, 20, 30 бөлімшелері үшін ғана).

SELECT сұрауларының базалық синтаксисі. Oracle SQL бар стандарт саласына сәйкес келеді. Дамып келе жатқан стандартпен бірге SQL бойынша стандарттарды жасау жөніндегі жұмыс комитеті Oracle корпорациясының белсенді қатысуының арқасында қамтамасыз етіледі. Мұндай жалпымойындалатын комитет болып Стандарттардың Америкалық Ұлттық

Институты (ISO) табылады. ANSI де, ISO да реляциондық деректер базасы үшін стандарт ретінде SQL қабылдаған.

SELECT операторында ANSI SQL салыстырғанда мынадай қосымша мүмкіндіктер бар:

- 1) NULL таңдау тізімінде.
- 2) Сұрау сұраудан (SELECT FROM (SELECT....)).
- 3) Оператордың IN сол бөлігі ANSI SQL бірлік сипатынан айырмашылығы тізімде бейнеленуі мүмкін.
- 4) Тек бағана ғана емес, сондай-ақ кез келген бейнелеу LIKE операторы арқылы пайдаланылуы мүмкін.
- 5) Тек жеке бағана ғана емес, кез келген өрнек, IS NULL және IS NOT NULL салыстырғанда оператор ретінде қолданылуы мүмкін.
- 6) ORDER BY сөйлемінде құрамында таңдау тізіміндегі бағаналардың тек қана аты, бүркеншек аты бойынша айырмашылық көрсететін кез келген FROM сөйлемі бар кез келген құрамы қолданылауы мүмкін.
- 7) GROUP BY сөйлемінде құрамында таңдау тізіміндегі бағаналардың тек қана аты, бүркеншек аты бойынша айырмашылық көрсететін кез келген FROM сөйлемі бар кез келген құрамы қолданылауы мүмкін.
- 8) Салынған агрегатты функциялар MIN(MAX(col1)) (салым деңгейі 2-ден көп емес).
- 9) Ішкі байланыс операторы (+).
- 10) Ағаш тәрізді сұраулар.
 - таңдау тізімінде NULL-мәні болады;
 - FROM сөйлемінде ішкі сұрау көрсетілген;
 - IN операторының сол жағында екі бағанадан тұратын тізім көрсетілген, ал оң жақта – екі бағананы кері қайтатын тізі тұр;
 - LIKE операторымен құрамында бағананың сілтемесі бар '%||ename2||%', сипаттама бейнеленген.
- 11) IS NOT NULL салыстыру операторымен sal+sal2 сипаты пайдаланылады.
- 12) Іріктеу sal+sal2 сипатының мәні бойынша жүзеге асады.

Oracle-дағы SELECT командасының базалық синтаксисі келесідей түрде болады:

```
SELECT [DISTINCT] список_столбцов  
FROM источник WHERE фильтр BY выражение_сортировки
```

LIKE операторын қолданудың ерекшеліктері. Метасимволдарды қиыстыру. «%» және «_» символдарын литеральды символдармен қиыстыруға болады. Слайдтағы мысал екінші әрібі – «А» болатын барлық қызметкерлердің тектерінің шығуын көрсетеді.

ESCAPE опциясы. Егер іздестіру метасимволдар арқылы емес, «%» және «_» нақты символдары жүргізілмесе, онда ESCAPE опциясын пайдалану қажет. Бұл опция символ береді, ол метасимволды қарапайым

символға айналдырады. Келесі SQL командасы құрамында «SA _» бар барлық қызметкерлерді тегін сұрайды.

```
SELECT employee_id, last_name, job_id  
FROM employees WHERE job_id LIKE '%SA\_%' ESCAPE '\';
```

ESCAPE опциясына кері қысық сызық белгісі (\) берілген. Шаблонда қысық сықыз астын сыз белгісімен (_) бірге келетін болғандықтан, Oracle сервері астын сызу белгісін метасимвол түрінде емес, әріп ретінде қабылдайды.

Мәліметтер қорынан мүлдем пайдаланылмайтын жолдардың сұрауларын орындау үшін, Oracle-да арнайы DUAL кестесі жасалған (SQL Server-де ол қарастырылмаған). Бұл кестеде – бір ғана DUMMY бағаны жалғыз х мәнімен. Кестенің осы қызметтік кестені пайдаланатын сұрау былай көрінуі мүмкін:

```
SELECT 10*15 FROM DUAL;
```

Операторлардың басымдылықтары. Бір пердиката логикалық операторлар келесі ретпен орындалады: алдымен NOT операторы орындалады, содан соң – AND және содан кейін ғана – OR операторы. Операторлардың орындалу ретін өзгерту үшін жақашарды қолдануға рұқсат беріледі.

AND операторының басымдылықтарының мысалы:

- 1) Біріншісі – AD_PRES мансабы және 15000 артық жалақы.
- 2) Екіншісі – SA_REP мансабы.

Осылайша, SELECT командасы былайша жазылады:

«Жолды таңдау, егер қызметкер – AD_PRES және \$15000 артық жалақы тапса немесе ол – SA_REP».

```
SELECT last_name, job_id, salary FROM employees WHERE  
job_id='SA_REP' OR job_id='AD_PRES' AND salary>15000
```

Жақшаларды пайдалану мысалы. Бұл мысалда екі шарт бар:

- 1) біріншісі – AD_PRES немесе SA_REP масабы,
- 2) екіншісі – 15000 артық жалақы.

Сәйкесінше, SELECT командасы келесі түрмен оқылады:

«Жоды таңдау, егер қызметкер – AD_PRES немесе SA_REP және ол \$15000 артық жалақы алса».

```
SELECT last_name, job_id, salary FROM employees WHERE  
(job_id='SA_REP' OR job_id='AD_PRES') AND salary>15000;
```

5 Зертханалық жұмыс № 5. SQL ДҚБЖ ORACLE-нің күрделі операторлары және көпжолдық функциялары

Жұмыстың мақсаты: ДҚБЖ ORACLE күрделі операторлары мен функцияларын игеру.

5.1 Зертханалық жұмысқа тапсырмалар

1. Бөлімшенің нөмірін, қызметкердің тегі мен берілген қызметкермен бірге бір бөлімде жұмыс жасайтын барлық қызметкерлердің тектерін енгізу үшін сұрау жасаңыз. Бағаналарға сәйкес атаулар беріңіз.
2. Zlotkey-мен бірге бір бөлімде жұмыс жасайтын қызметкерлердің әрқайсысы үшін оның тегі және жалдану датасын енгізу үшін сұрау құрыңыз. Шығу деректерінің Zlotkey-ді алып тастаңыз.
3. Бөлімшесі `location_ID = 1700` жатқан әрбір қызметкердің тегін, бөлімше нөмірін және әрбір қызметкерді енгізу үшін сұрау жасаңыз.
4. Әкімшілік департаментте әрбір жұмыс істейтін қызметкердің тегін, бөлімше нөмірін және әрбір қызметкердің мансабын алыңыз (Executive).
5. Бөлімшенің нөмірін, қызметкердің тегі мен берілген қызметкермен бірге бір бөлімде жұмыс жасайтын барлық қызметкерлердің тектерін енгізу үшін сұрау жасаңыз. Бағаналарға сәйкес атаулар беріңіз.
6. Бөлімшесі `location_ID = 1700` жатқан әрбір қызметкердің тегін, бөлімше нөмірін және әрбір қызметкерді енгізу үшін сұрау жасаңыз.
7. Kochhar секілді комиссиялық шығын алатын қызметкердің тегін, бөлімше нөмірін және әрбір қызметкердің мансабын енгізу үшін сұрау жасаңыз.
8. Ескерту: нәтижелік көптікте Kochhar көрсетілмеуі керек .
9. T-дан басталатын, қалаларды өмір сүретін қызметкерлер туралы деректер енгізіңіз (қызметкердің нөмірі, тегі мен бөлімшенің нөмірі).
10. NOT EXISTS операторын пайдалана отырып жетекшілері жоқ барлық қызметкерерді табыңыз
11. MY_EMPLOYEE кестесіне жолдарды интерактивті енгізу үшін `loademp.sql` скрипт-файлын құрыңыз. Берілген қызметкер үшін пайдаланушы идентификаторын беру үшін аттың бірінші әріпін жеті тектің бірінші әріпімен біріктіріңіз.
12. Құрамында қызметкердің нөмірі, тегі мен 50 бөлімшесінің барлық қызметкерлерінің нөмірі бар DEPT50 табыстауын құрыңыз. Табыстаудың бағаналарын EMPNO, EMPLOYEE, DEPTNO деп атаңыз. Табыстау арқылы басқа бөлімшелердегі (50-ден басқасы) қызметкерлердің алмасу операциясына тиым салыңыз.

Бақылау сұрақтары:

- 1) Бірнеше кестеден деректерді таңдау қалай жүргізіледі?
- 2) Байланыстың қандай түрлері бар?
- 3) SELECT инструкциясында қандай функция бір сұрау блогын көпрет қолдануға мүмкіндік береді?
- 4) MERGE операторы е үшін қолданылады?
- 5) Көбейтурегген не үшін операциялар қолданылады?

5.2 Зертханалық жұмысты орындауға арналған әдістемелік нұсқаулар

Жұмыстың орындалу реті.

1) Ішкі сұраулар.

a) орташа жалақы мөлшерінен аз жалақы алатын барлық қызметкерлер аты-жөнін енгізіңіз.

```
SELECT last_name, job_id, salary
FROM employees
WHERE salary < (SELECT AVG(salary) FROM employees)
```

b) 141 номерінде қызмет атқаратын қызметкерлерді енгізіңіз және 143 номерінде қызмет атқаратындардан оклады көп қызметкерледі енгізіңіз. Мысал сұраудың үш блогынан тұрады: бір сыртқы және екі ішкі. Алдымен ішкі блоктарды орындап алған дұрыс. Олардың нәтижелері ST_CLERK және 2600. Содан соң сыртқа блоктарды орындау керек. екі ішкі блокта бір мәнге ие болады ST_CLERK және 2600, сондықтанда ол команда біржолақты подзапрос деп аталыд.

Ескерту: ішкі және сыртқы сұйық сұраулар деректерді әр түрлі кестелерден таңдай алады.

```
SELECT last_name, job_id, salary
FROM employees
WHERE job_id =
      (SELECT job_id
       FROM employees
       WHERE employee_id = 141)
AND salary >
      (SELECT salary
       FROM employees
       WHERE employee_id = 143)
```

c) hr.employees кестесінің ақпараттары негізінде қызметкерлердің аттары мен тегін қайтаратын сұрау жазыңыз. Жазбалар тек менеджерлік функция атқаратын қызметкерлер үшін ғана қайтуы керек (олардың нөмірлері бір кестенің employee_id сол manager_id бағанында болатын). Осы тапсырманы көпжолды ішкі сұрау арқылы шешіңіз.

```
SELECT first_name As "Имя", last_name As "Фамилия"
FROM hr.employees
WHERE employee_id = ANY (SELECT manager_id
                        FROM hr.employees)
```

d) Коррелирленген ішкі сұраулар. Жұмыс орындары минимум екі рет ауыстырылған қызметкерлердің деректерін көрсетіңіз.

```
SELECT e.employee_id, last_name, e.job_id
FROM employees e
WHERE 2 <= (SELECT COUNT(*)
           FROM job_history
           WHERE employee_id = e.employee_id)
```

2) Байланыс.

Қызметкерлердің тізімі мен олар жұмыс істейтін бөлімнің атауларын жазыңыз.

```
SELECT e.employee_id, e.last_name, e.department_id, d.department_id, d.location_id
FROM employees e JOIN departments d
ON e.department_id = d.department_id
WHERE last_name = 'Matos'
```

3) MERGE операторы арқылы сұрауларды орындау. Жолдардың үйлесуін орындау - COPY_EMP кестесінде EMPLOYEE_ID сәйкес және EMPLOYEES кестесінде EMPLOYEE_ID сәйкес жүред. Егер сәйкестік табылса, COPY_EMP кестесінің жолы EMPLOYEES кестесінің жолына сәйкес модификацияланады. Егер жол табылмаса, ол COPY_EMP кестесіне қойылады.

```
MERGE INTO copy_emp AS c
USING employees e
ON (c.employee_id = e.employee_id)
WHEN MATCHED THEN
UPDATE SET
c.first_name = e.first_name, c.last_name = e.last_name,
c.email = e.email, c.phone_number = e.phone_number,
c.hire_date = e.hire_date, c.job_id = e.job_id,
c.salary = e.salary, c.commission_pct = e.commission_pct,
c.manager_id = e.manager_id, c.department_id = e.department_id
WHEN NOT MATCHED THEN
INSERT VALUES(e.employee_id, e.first_name, e.last_name,
e.email, e.phone_number, e.hire_date, e.job_id,
e.salary, e.commission_pct, e.manager_id, e.department_id)
```

Ішкі сұраулар.

Ішкі сұраулар – бұл басқа команданың SELECT сөйлемінде қосылған, SELECT командасы. Ішкі сұраулардың көмегімен қарапайым командалардан өте мықты командалар жасауға болады. Олар, егер кестедегі жолдарды таңдау кестенің өзіндегі деректерге тәуелді шарт бойынша жүргізілетін жағдайларда өте пайдалы.

Ішкі сұрауды SQL тілінің WHERE, HAVING және FROM сөйлемдерінде пайдалануға болады.

Синтаксис:

operator салыстыру операторы – мысалы, > , = немесе IN

Ішкі сұрауды көбіне SELECT ішкі командасына қосылған SELECT командасының салындысы деп атайды. Әдетте ішкі сұрау бірінші болып орындалады, және де оның нәтижесі басты немес ішкі сұрауды шартты анықтау үшін пайдаланылады.

Ішкі сұрауларды пайдалануға қатысты көрсетілімдер:

1) Ішкі сұраулар жақшаға алыну керек.

2) Ішкі сұраулар теңестіру операторының оң жағында жатуы керек.

3) Oracle8i релизіне дейін ішкі сұрауларда ORDER BY сөйлемі болмаған. Егер ORDER BY сөйлемі пайдаланылған болса, онда ол SELECT басты командасында соңғы болуы керек еді. Oracle8i релизінен бастап, ORDER BY көрсеткіші ішкі сұрауларда *top-n* талдауды орындау үшін қолданылуы мүмкін.

6 Зертханалық жұмыс №6. Oracle функциясының қосымша мүмкіндіктері

Жұмыстың мақсаты: аналитикалық функцияларды зерттеу.

6.1 Зертханалық жұмысқа тапсырмалар

1. Ертеректе барлығы жұмысқа тұрған қызметкерлердің алғашқы үшеуін енгізу.

2. Қол астында жұмыс істейтін адамдардың саны максималды қызметкерді табу.

3. Бағынушылардың департаментін вице президентке шығару.

4. Барлық сату жылына – «олардың» сатып алушыларының сату суммалары бойынша сатып алушыларғы қызмет көрсететіндердің (customer кестесіндегі salesperson_id) компания қызметкерлерінің рангін орнататын сұрау жазу. Сатудың максималды суммасына «1» рангі сәйкес келеді, келесі максималдыға – «2» рангі және т.б. Егер сату суммасы бірдей болса, онда ранг те бірдей болуы керек. ОЕ сұлбасын пайдалану.

5. Бөлімшенің нөмірін, қызметтік дәреженің кодын, осы екі бағана бойынша топтастырылған жалақы суммасын, сонымен бірге аралық қорытындыларды, бөлімшелер бойынша жалақының ортақ суммасын есептейтін сұраулар жазу.

Бақылау сұрақтары:

1) Аналитикалық функциялар не үшін қолданылады?

2) Псевдокестелер қалай жұмыс жасайды?

3) Ағаштәрізді сұраулар қандай қатар бойынша құрылады?

4) Терезелік функциялардың қандай түрлерін білесіз?

5) Топтастыруда қандай функциялар пайдаланылады?

6.2 Зертханалық жұмысты орындауға арналған әдістемелік нұсқаулар

Жұмысты орындау реті.

1) Аналитикалық функциялар. Ең жас адамдардың *top - 10* тізімін алыңыз:

```
SELECT RANK() OVER (ORDER BY HIRE_DATE ASC) AS ranking,
```



```
employee_id, last_name, hire_date
FROM employees WHERE ranking <= 10
```

Аталмыш код 10 қатардан артық қайтуы мүмкін. Мысалы, егер жасы бірдей екі адам бар болса, онда 11 тәулікте қайтады.

2) Деректерді топтасытыру

Қызмет мансабы, департаменттер бойынша жалақыны есептеңіз және топтастырыңыз, ортша жалақыны есептеңіз.

```
SELECT department_id, job_id, manager_id, SUM(salary)
FROM employees
GROUP BY department_id,
        ROLLUP (job_id),
        CUBE (manager_id)
```

3) Рекурсивті сұраулар.

Қызметкерлердің қандай саны жыл сайын 2002 жыл мен 2010 жыл аралығында түсетінін есептеңіз.

```
WITH period ( year ) AS (SELECT 2002 AS year FROM dual
UNION ALL
SELECT year + 1 AS year FROM period WHERE year < 2010)
SELECT p.year, COUNT ( e.employee_id )
FROM employees e RIGHT OUTER JOIN period p
ON p.year = EXTRACT ( YEAR FROM e.hire_date )
GROUP BY p.year ORDER BY p.year
```

4) Ағаштәрізді сұраулар.

Менеджерлер мен қызметкерлердің тізімін енгізіңіз әзірше employee_id=101.

```
SELECT employee_id, last_name, job_id, manager_id
FROM employees START WITH employee_id = 101
CONNECT BY PRIOR manager_id = employee_id
```

Аналитикалық функциялар. Псевдобаған.

Oracle-дағы псевдобағандар деп кестеде анық көрінбеген, алайда сұрауда қолданылуы мүмкін бағандар.

ROWID псевдобағаны — жолдың керемет идентификаторы болып табылады. Ол тек қана кесте аясында ғана керемет емес, сондай-ақ деректер қорында да керемет болып табылады. **ROWID** физикалық көзғараста қордағы өзіндік координатты жазба болып табылады.

ROWID деректер қорымен жұмысты жеңілдетеді, кез-келген кесте бағанын бірденнен идентифицирлей алады, алғашқы кілтсіз кесте бағанын жояа және түзете алады. Сонымен қатар, **ROWID** қосымшаға ең тез әсер ететін және деректерді белсенде модифицирлейтді.

Алайда **ROWID** Oracle арнайы ерекшелігі болғандықтан, оны қосымшалар жасауда қолдануға болмайды.

```
SELECT ROWID, Last_Name FROM hr.employees
```

ROWNUM псевдобағаны ORACLE SQL сұрауындағы деректерді шектеуге қолданады. Oracle нәтижелері қайталанытн жолдарды нөмерлей алады. Бұл үшін арнаы *ROWNUM псевдобағаны* қолданылады. Алайда бұл бағанды тура сұрауда қолдану мүмкін емес: мыслы, мынадай сұрау:

```
SELECT ROWNUM, Last_Name FROM hr.employees
```

Қатені қайтарады.

Мұнымен бірге ROWNUM шығарылатын жазбалар санын шектеуге қолданылады:

```
SELECT ROWNUM, Last_Name FROM employees WHERE ROWNUM < 10
```

Мұндай жағдайда барлығы 9 жазба беріген.

Мұндай мүмкіндік сұрау мәндері өте үлкен болған кезде жне серверді қатты жүктеген кезде сізге тек қана қалыпты жұмыс істеп жатқанына көз жеткізу үшін керек. аналитикалық функцияларды енгізудің техникалық мақсаты – лаконикалық формулировка беру және «аналитикалық сұрау» жылдамдығын арттыру.

Top-n Analysis орындау. Top-N анализды подзапроса көмегімен және ішкі запросар арқылы орындауға болады.

Top-n сұрау төменде сипатталған жағдайларда қолданылады:

1) Ішкісұраныс (inline view) сұрыпталған деректер тізімін генерациялау үшін. подзапрос деректерді қажетті ретпен қойюды қамтамасыз ету үшін ORDER BY қосады. Нәтижелерді кему реті бойынша қою үшін DESC параметрі керек.

2) Сыртқы сұрау нәтижелік жиынтықта баған санын шектейді. Сыртқы сұрау келесілерден тұрады:

– псевдобаған ROWNUM, кезектес мәнді көрсетеді, подзапростың әрбір қайталанытын бағаны 1ден басталады;

– WHERE, қайталану үшін n бағанын береді. Ішкі WHERE –де (<) немесе (<=) операторын қолдану керек.

Top-N анализын жүргізудің жалпы көрінісі:

```
SELECT [список_столбцов], ROWNUM
FROM (SELECT [список_столбцов] FROM таблица
ORDER BY Top-N_столбец) WHERE ROWNUM <= N
```

Агрегирлеудің қарапайым функциясымен салыстыру:

Сұраныстың әрбір бағаны басқа бағанға бағынышсыз өңделеді. Алайда, іс жүзінде бағанды өңдеу барысында оны алдыңғы немесе келесі бағанмен салытыру керек, бағандар тобын белгілеу керек, бір-біріне бағынышсыздарын өңдеу және т.б. ДҚБЖ SQL тілінде іш жүзінде қолданылуы сай келетін конструкция негізінде пайда болған, әдетте сап түзеу мен аналитикалық функциялар SQL:2003 тіркелген.

ДҚБЖ Oracle агрегатты функциялары (max(), min(), avg()) SQL талаптарына сай жұмыс жасайды. Агрегаттық және аналитикалық функциялардың жалпы саны Oracle11g-да 50ден астам. Аналитикалық

функциялар сап түзеуге (ranking), нәтижелерді ауыстыруға (moving aggregates), деректерлі түрлі период бойынша салыстыруға (period comparisons), қорытынды қатынасын көрсетуге (ratio of total), және т.б. әрекеттерді орындауға мүмкіндік береді.

Көптеген аналитикалық функциялар карапайым скалярлы агрегирленген SUM, MAX функциялары сияқты әрекет етеді және GROUP BY арқылы қалыптасады.

Аналитикалық функция жалпы синтаксисте қолданылады және арнайы мүмкіндіктерге ие. Аналитикалық функция принциптерін түсіну үшін төмендегілерді түсіну керек:

- деректерді қай скенсирлеу керек;
- деректерді қалай реттеу;
- деректер терезесін қалай беру.

Барлық функциялар келесі синтаксисте ие:

ИМЯ_ФУНКЦИИ(<аргумент>,< аргумент >, . . .)

OVER

(<конструкция_фрагментации> <конструкция_упорядочения> <конструкция_окна>)

OVER — бұл функцияны аналитикалық ретінде идентифицирлейтін кілттік сөз. Жоғарыда SUM агрегирлеуші функциясы көрсетілген:

SUM(выражение 1) OVER([PARTITION BY выражение 2 [, выражение 3 [, ...]]])

Егер де PARTITION BY көрсетілмесе, онда жалғыз топ есебінде толық мәтін алынады:

```
SELECT ename, deptno, job, SUM(sal) OVER () sum_sal FROM emp
```

Аналитикалық функциялар мүмкіндіктеріне қарай төрт түрлі сыныпқа бөлінеді:

1) Сап түзеу функциясы, «бірінші N» типті сұранысын құруға мүмкіндік береді.

2) Терезелік функциялар, түрлі агрегаттарды шығаруға мүмкіндік береді.

3) Қорытынды функциялар. Қорытынды функциядан терезелі функцияның кілттік айырмашылы – OVER операторындағы ORDER BY құрылымының болмауы. ORDER BY құрылымы болмаған кезде функция әрбір топ бағанында қолданылады. ORDER BY құрылымы бар кезде функция терезеде қолданылады.

4) VAR_POP, VAR_SAMP, STDEVJPOP сияқты статистикалық функциялар, сызықтық регрессивті функциялар жиынтығы және т.б. бұлар кез келген ретсіз секцияның статистикалық көрсеткішін шығаруға мүмкіндік береді.

Бір аналитикалық функцияны басқасында қолдануға болмайды.

```
SELECT KEEP (DENSE_RANK FIRST ORDER BY [ NULLS )  
OVER (PARTITION BY ) FROM GROUP BY
```

Әдебиеттер тізімі

1. Сатимова Е.Г. Oracle: особенности SQL. Лабораторный практикум. – Алматы, 2013.
2. Аруп Нанда и Стивен Фейерштейн, Oracle PL/SQL для администраторов баз данных. - Символ-Плюс. 2008. - 494 стр
3. Джеймс Перри, Джеральд Пост, Введение в Oracle 10g. - Вильямс. 2006. - 697 стр.
4. Урман С., Oracle Database 10g. Программирование на языке PL/SQL. - М.: «Лори». 2010.
5. Прайс Д., Oracle 10g SQL. - М.: «Лори». 2010.
6. Нанда А. , Oracle PL/SQL для администраторов баз данных. - СПб.: «Символ».- 2008.
7. Каучмэн Д., Oracle Certified Professional DBA: подготовка администраторов баз данных. - М.: «Лори». 2009.
8. Браун Б., Oracle Database. Создание Web-приложений. - М.: «Лори». 2010.

Аманбаев Абдирахман Абдиханович

Сатимова Елена Григорьевна

ORACLE ДЕРЕКТЕР ҚОРЫН БАСҚАРУ ЖҮЙЕЛЕРІ

1 БӨЛІМ

5B070400 - Есептеу техникасы және бағдарламалық камтамасыз ету мамандығының студенттері үшін зертханалық жұмыстарды орындау бойынша әдістемелік нұсқаулықтар

Редактор: Б.С. Касимжанова

Стандарттау бойынша маман Н.К. Молдабекова

Басылымға жіберілді ____ . ____ . ____

Пішімі 60x84 1/16

Таралымы 50 дана.

№1 типографиялық қағаз

Көлемі 2,8 оқу-бас.ә.

Тапсырыс ____ Бағасы 1400 тг.

«Алматы энергетика және байланыс университеті»

коммерциялық емес акционерлік қоғамының

көшірмелік-көбейткіш бюросы

050013, Алматы, Байтурсынулы, 126