**DBMS Lab Assignment – 1**

Date: 30th August, 2023

* **Steps to create a user in oracle:**

**Step 1:** Open SQL Plus tool (Open command prompt, Type sqlplus)

**Step 2:** Login to the Database.

user name: system

password: abcd1988

**Step 3:** Alter session with the following command:

SQL> alter session set "\_oracle\_script “=true;

**Step 4:** Create user with the following command:

SQL> create user username identified by password;

**Step 5:** grant privileges to the created user “username”:

SQL> grant all privileges to username;

* **If you have already created the user, then connect to the user using the following syntax:**

**Step 1:** Open SQL Plus tool (Open command prompt, Type sqlplus)

**Step 2:** Login to the Database.

user name: system

password: abcd1988

**Step 3:** Login to the user using the following syntax

SQL> conn

**Step 4:** Enter the user name and password that you created.

**1. Create the following tables in the Relation database:**

**Syntax: CREATE TABLE table\_name (fieldname1 datatype1, fieldname2 datatype2, …, constraints(fieldname)); [Please note: Constraints are optional]**

**Create the student table**

SQL>create table student (snum numeric(9,0), sname varchar(15), major varchar (20), standing varchar(2), age numeric(3,0), primary key (snum));

**Create the faculty table**

SQL>create table faculty(fid numeric(9,0),fname varchar(25),deptid numeric(2,0),primary key(fid));

**Create class table**

SQL>create table class (name varchar (30), meets\_at varchar (20), room varchar (10), fid numeric (9,0), primary key (name), foreign key(fid) references faculty(fid));

**Create enrolled table**

SQL>create table enrolled(snum numeric(9,0),cname varchar(40),primary key(snum,cname), foreign key(snum) references student(snum),foreign key(cname) references class(name));

**2. Check the tables that you created: (this command displays all the tables under this user)**

**Syntax: SELECT \* FROM TAB;**

SQL>select \* from tab;

**3. Check description of individual tables:**

**Syntax: DESC table\_name;**

SQL> desc student;

SQL> desc faculty;

SQL> desc class;

SQL> desc enrolled;

**4. Insert the following values into student table using the insert statement.**

**Syntax: INSERT INTO table\_name VALUES (123, ‘xyz’, …);**

Example: Insert values into student table

SQL> INSERT INTO student VALUES(051135593,'Maria White','English','SR',21);

**Similarly insert the rest of the data into student table so that the table contains the following data:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **snum** | **sname** | **major** | **standing** | **age** |
| 05593 | Maria White | English | SR | 21 |
| 06453 | Charles Harris | Architecture | SR | 22 |
| 09543 | Susan Martin | Law | JR | 20 |
| 11546 | Joseph Thompson | Computer Science | SO | 19 |
| 11938 | Christopher Garcia | Computer Science | JR | 20 |
| 13562 | Angela Martinez | History | SR | 20 |
| 26834 | Thomas Robinson | Psychology | SO | 18 |
| 28572 | Margaret Clark | Animal Science | FR | 18 |
| 30823 | Juan Rodriguez | Psychology | JR | 20 |
| 31912 | Dorthy Lewis | Finance | FR | 18 |
| 32981 | Daniel Lee | Electrical Engineering | FR | 17 |
| 32189 | Lisa Walker | Computer Science | SO | 17 |
| 34549 | Paul Hall | Computer Science | JR | 18 |
| 35322 | Nancy Allen | Accounting | JR | 19 |
| 45864 | Mark Young | Finance | FR | 18 |
| 45411 | Luis Hernandez | Electrical Engineering | FR | 17 |
| 46489 | Donald King | Mechanical Engineering | SO | 19 |
| 55548 | George Wright | Education | SR | 21 |
| 55318 | Ana Lopez | Computer Engineering | SR | 19 |
| 55565 | Kenneth Hill | Civil Engineering | SR | 21 |
| 56612 | Karen Scott | Computer Engineering | FR | 18 |
| 57895 | Steven Green | Kinesiology | SO | 19 |
| 57456 | Betty Adams | Economics | JR | 20 |
| 57478 | Edward Baker | Veterinary Medicine | SR | 21 |

**FACULTY**

Example: Insert values into faculty table

SQL> INSERT INTO faculty VALUES (1464,'Ivana Teach',20);

**Similarly insert the rest of the rows into the table so that the final table looks as below:**

|  |  |  |
| --- | --- | --- |
| **fid** | **fname** | **deptid** |
| 1464 | Ivana Teach | 20 |
| 2465 | James Smith | 68 |
| 1451 | Mary Johnson | 20 |
| 0112 | John Williams | 68 |
| 2523 | Patricia Jones | 68 |
| 3525 | Robert Brown | 12 |
| 4822 | Linda Davis | 20 |
| 2812 | Michael Miller | 12 |
| 2455 | Barbara Wilson | 12 |
| 1516 | William Moore | 33 |
| 0919 | Elizabeth Taylor | 11 |
| 4866 | David Anderson | 20 |
| 6188 | Jennifer Thomas | 11 |
| 4823 | Richard Jackson | 33 |
| 5462 | Ulysses Teach | 20 |
| 1234 | Mr. Jackson | 20 |

**CLASS**

Example: Insert values into class table

SQL> INSERT INTO class VALUES (‘Data Structures’,’MWF 10’, ‘R128’, 4822);

**Similarly insert the rest of the rows into the table so that the final table looks as below:**

|  |  |  |  |
| --- | --- | --- | --- |
| **NAME** | **MEETS\_AT** | **ROOM** | **FID** |
| Data Structures | MWF 10 | R128 | 4822 |
| Database Systems | MWF 12:30-1:45 | 1320 DCL | 1464 |
| Operating System Design | TuTh 12-1:20 | 20 AVW | 4822 |
| Archaeology of the Incas | MWF 3-4:15 | R128 | 2455 |
| Aviation Accident Investigation | TuTh 1-2:50 | Q3 | 0112 |
| Air Quality Engineering | TuTh 10:30-11:45 | R15 | 0112 |
| Introductory Latin | MWF 3-4:15 | R12 | 2455 |
| American Political Parties | TuTh 2-3:15 | 20 AVW | 6188 |
| Social Cognition | Tu 6:30-8:40 | R15 | 1516 |
| Perception | MTuWTh 3 | Q3 | 4823 |
| Multivariate Analysis | TuTh 2-3:15 | R15 | 0919 |
| Patent Law | F 1-2:50 | R128 | 0919 |
| Urban Economics | MWF 11 | 20 AVW | 4823 |
| Organic Chemistry | TuTh 12:30-1:45 | R12 | 4823 |
| Marketing Research | MW 10-11:15 | 1320 DCL | 4823 |
| Seminar in American Art | M 4 | R15 | 4823 |
| Orbital Mechanics | MWF 8 | 1320 DCL | 0112 |
| Dairy Herd Management | TuTh 12:30-1:45 | R128 | 3525 |
| Communication Networks | MW 9:30-10:45 | 20 AVW | 1451 |
| Optical Electronics | TuTh 12:30-1:45 | R15 | 2523 |
| Intoduction to Math | TuTh 8-9:30 | R128 | 4823 |

**ENROLLED**

Example: Insert values into enrolled table

SQL> INSERT INTO enrolled VALUES(11546, ‘Database Systems’);

**Similarly insert the rest of the rows into the table so that the final table looks as below:**

|  |  |
| --- | --- |
| **SNUM** | **CNAME** |
| 11546 | Database Systems |
| 11938 | Database Systems |
| 34549 | Database Systems |
| 32189 | Database Systems |
| 55318 | Database Systems |
| 45411 | Operating System Design |
| 55318 | Operating System Design |
| 56612 | Operating System Design |
| 11546 | Operating System Design |
| 11938 | Operating System Design |
| 32189 | Operating System Design |
| 56612 | Data Structures |
| 55318 | Communication Networks |
| 45411 | Optical Electronics |
| 30823 | Perception |
| 30823 | Social Cognition |
| 30823 | American Political Parties |
| 55565 | Air Quality Engineering |
| 09543 | Patent Law |
| 57456 | Urban Economics |

**5. Verify if the data was successfully added to the table:**

**Syntax: SELECT \* FROM table\_name;**

SQL> select \* from student;

SQL> select \* from faculty;

SQL> select \* from class;

SQL> select \* from enrolled;

**6. Delete data from table:**

**Syntax: DELETE FROM table\_name WHERE condition; (condition can be equality, inequality and so on).**

Example: Mr. Jackson is a temporary, substitute teacher. We need to remove rows pertaining to his records from the faculty table

SQL> DELETE FROM faculty WHERE FID = 1234;

**7. Some exercises on altering tables**

**Alter Table- RENAME column**

**Syntax: ALTER TABLE table\_name RENAME COLUMN column\_name TO new\_col\_name;**

**Example:** The column name ‘fname’ in faculty table is not very intuitive. Rename the column name to faculty name:

SQL> ALTER TABLE faculty RENAME COLUMN fname TO facultyName;

**Alter Table- ADD column**

**Syntax: ALTER TABLE table\_name ADD column\_name DATATYPE;**

Example: Write the SQL query toadd a column **telephone** of datatype integer of size 10 to the student table, then add a column roll no. of type numeric(5)

SQL**>** ALTER TABLE student ADD telephone numeric(10);

SQL**>** ALTER TABLE student ADD roll\_no numeric(10);

**Alter Table- DROP column**

**Syntax: ALTER TABLE table\_name DROP COLUMN column\_name;**

Example: Write the SQL query toremove a column **telephone** from the student table.

SQL**>** ALTER TABLE student DROP COLUMN telephone;

**7. Exercise on Renaming tables**

**Syntax: ALTER TABLE table\_name RENAME TO new\_table\_name;**

**Example:** Rename the table enrolled to enrollment:

SQL> ALTER TABLE enrolled RENAME TO enrollment;

**8. Exercise on Updating records in table**

**Syntax: UPDATE table\_name SET column\_name = value WHERE condition; Note: condition can be equality or inequality against any column name in table**

**Example:** Update age of ‘Maria White’ to 23 in student table

SQL> UPDATE student SET age=23 WHERE sname='Maria White';

**9. Exercise on Deleting records in table**

**Delete all records**

**Syntax: DELETE FROM table\_name;**

Example: Write the SQL query to delete all rows of enrollment table without deleting the table.

SQL> DELETE FROM enrollment;

**Delete Specific records**

**Syntax: DELETE FROM table\_name WHERE condition;**

Example: Write the SQL query to delete the student details whose sname is “Maria White”.

SQL> DELETE FROM student WHERE sname = ‘Maria White’;

**10. Exercise on Truncate a table**

**Truncate command is used to delete all rows from a table.**

**Syntax: TRUNCATE TABLE <TABLE NAME>;**

Example: Write the query to truncate table enrollment

SQL> TRUNCATE TABLE enrolled;

**11. Exercise on Dropping a table**

**Use the Drop Statement to Remove a table**

**Syntax: DROP TABLE table\_name;**

Example: Write the query to drop table enrollment

SQL> DROP TABLE enrolled;

**12. Exercise on Recyclebin**

* It is a system-defined table that is used for storing the information about the dropped tables.
* To view the structure of recycle bin, you can use the following syntax:

SQL> DESC RECYCLEBIN;

* To view dropped tables in recycle bin, you can use the following syntax:

SQL> SELECT ORIGINAL\_NAME, OPERATION, TYPE FROM RECYCLEBIN;

**13. Exercise on flashback command**

* The flashback command is used to restore the dropped table from recycle bin to the database.
* The syntax to use the Flashback command is given below.

Syntax: FLASHBACK TABLE <TABLE NAME> TO BEFORE DROP;

* Example: SQL> flashback table employee to before drop;

**14. Exercise on purge command**

* purge command is used to drop a table permanently i.e. it will drop a specific table from recycle bin permanently.
* The syntax to use the purge command is given below.

Syntax: PURGE TABLE <TABLE NAME;

**15. Execute the following sequence of commands and understand the difference between Truncate, Drop, and Purge.**

SQL> Select \*from class;

SQL> Truncate table class;

SQL> Select \*from class;

SQL> Drop table class;

SQL> Select \*from class;

SQL> Flashback table class to before drop;

SQL> Select \*from class;

SQL> Drop table class;

SQL> SELECT ORIGINAL\_NAME, OPERATION, TYPE FROM RECYCLEBIN;

SQL>Purge table class;

SQL> SELECT ORIGINAL\_NAME, OPERATION, TYPE FROM RECYCLEBIN;