Islamic University of Technology

Database Management Systems Lab

CSE 4308 / CSE 4174

Lab 1

Author:
Ishmam Tashdeed
Zannatun Naim Sristy
Lecturer
CSE, IUT

Contents

1 Data Definition

1.1 Creating a Table

The general syntax for creating a table is as follows:

```
CREATE TABLE table_name
(
 attribute1 datatype [ NULL | NOT NULL | UNIQUE],
 attribute2 datatype [ NULL | NOT NULL | UNIQUE],
 ...
);
```

There exists different data types in Oracle. Some of them are as follows:

- `char(n)`: value contains exactly n alpha-numeric characters
- `varchar2(n)`: value contains at most n alpha-numeric characters
- `number`: any integer or real numbers
- `date`: DD-MMM-YY format, like '20-JAN-22'

Assume that you want to create a table named '`CITIZEN`' with 3 attributes:

1. `national_id`: `number` type
2. `name`: `varchar2` type
3. `birth_date`: `date` type

To create this table, you have to write the following query:

```
CREATE TABLE CITIZEN
(
 NATIONAL_ID NUMBER NOT NULL,
 NAME VARCHAR2(50) NOT NULL,
 BIRTH_DATE DATE
);
```

To create a table with constraints, the syntax is given below:

```
CREATE TABLE table_name
(
 attribute1 datatype [ NULL | NOT NULL | UNIQUE],
 attribute2 datatype [ NULL | NOT NULL | UNIQUE],
 ...,
 [CONSTRAINT constraint_name] PRIMARY KEY (primary_attribute1, ...),
 [CONSTRAINT constraint_name] CHECK condition
);
```

Primary key is a special column that is able to uniquely identify each record. For example, if we want to create a table called CITIZEN, it might have the NATIONAL_ID as the primary key.

```
CREATE TABLE CITIZEN
(
    NATIONAL_ID NUMBER NOT NULL,
    NAME VARCHAR2(50) NOT NULL,
    AGE INT,
    COUNTRY VARCHAR2(20),
    CONSTRAINT PK_CITIZEN PRIMARY KEY(NATIONAL_ID),
    CONSTRAINT AGE_CHECK CHECK (AGE > 17 AND COUNTRY='BANGLADESH')
);
```

1.2 Dropping Tables

To delete the table, we use:

```
DROP TABLE table_name;
```

It will also delete the information stored in the table. To delete the table structure with constraints, we generally follow this command:

```
DROP TABLE table_name CASCADE CONSTRAINTS;
```

An example could be:

```
DROP TABLE CITIZEN CASCADE CONSTRAINTS;
```

1.3 Altering Tables

To add a new attribute to the table, we use:

```
ALTER TABLE table_name ADD attribute_name datatype;
```

We can even add multiple attributes at the same time:

```
ALTER TABLE table_name ADD (attribute1 datatype, ...);
```

An example could be:

```
ALTER TABLE CITIZEN ADD PHONE_NO VARCHAR2(12);
```

To delete an attribute from a table, we use:

```
ALTER TABLE table_name DROP COLUMN attribute_name;
```

We can even delete multiple attributes at the same time:

```
ALTER TABLE table_name DROP COLUMN (attribute1, ...);
```

Such as:

```
ALTER TABLE CITIZEN DROP COLUMN PHONE_NO;
```

To modify the data type of an attribute, we need to ensure that the column is **empty**. Then we can execute:

```
ALTER TABLE table_name MODIFY attribute_name new_datatype;
```

Such as:

```
ALTER TABLE CITIZEN MODIFY AGE FLOAT(2, 1);
```

To rename an attribute, we use:

```
ALTER TABLE table_name RENAME COLUMN old_attribute_name TO new_attribute_name;
```

To rename a table, we use:

```
ALTER TABLE table_name RENAME TO new_table_name;
```

2 Data Manipulation

2.1 Inserting Records/Rows into a Table

The general format for inserting a new record is:

```
INSERT INTO TABLE_NAME VALUES (..., ..., ...);
```

We can type the following command to insert new records into our 'CITIZEN' table using positional notation:

```
INSERT INTO CITIZEN VALUES (2015001, 'W', 19, 'Bangladesh');
```

We can insert new records into 'CITIZEN' table following named notation:

```
INSERT INTO CITIZEN(NATIONAL_ID, NAME, AGE, COUNTRY) VALUES
(2015002, 'X', 23, 'Bangladesh');
```

2.2 Updating existing data

The UPDATE statement is used to modify the existing records in a table.

```
UPDATE TABLE_NAME
SET column1 = value1, column2 = value2, ...
WHERE <CONDITIONAL CLAUSE>;
```

An example could be:

```
UPDATE CITIZEN
SET NAME = 'Hisenberg', COUNTRY= 'USA'
WHERE NATIONAL_ID = 1;
```

2.3 Deleting data from table

The DELETE statement is used to delete the existing records in a table.

```
DELETE FROM TABLE_NAME
WHERE <CONDITIONAL CLAUSE>;
```

An example could be:

```
DELETE FROM CITIZEN
WHERE NATIONAL_ID = 1;
```

2.4 Retrieval of Information

The basic SQL syntax of a query is as follows:

```
SELECT ATTRIBUTE1, ATTRIBUTE2
FROM TABLE_NAME
WHERE <CONDITIONAL CLAUSE>;
```

For example, to see the 'national_id' of all citizens from the 'CITIZEN' table:

```
SELECT NATIONAL_ID FROM CITIZEN;
```

To find the information of a citizen following a condition:

```
SELECT NATIONAL_ID FROM CITIZEN WHERE ID = 2015001;
```

To find the information of a citizen following multiple conditions:

```
SELECT NATIONAL_ID FROM CITIZEN WHERE AGE>21 AND
COUNTRY='Bangladesh';
```

```
SELECT NATIONAL_ID FROM CITIZEN WHERE AGE>21 OR NAME='W';
```

To find all records and their information:

```
SELECT * FROM CITIZEN;
```

3 Executing SQL Script in SQLPlus

You can create a file with `.sql` extension that contains your SQL statements. Then you can execute it from the SQLPlus command line directly.

Suppose, you have written your SQL statements in a file `a.sql` saved under `d:\sample\` directory. To execute that script, you have to type after logging in to your account:

```
@d:\sample\a.sql
```

4 Lab Task

You have to write all SQL statements in an editor first and save them with `.sql` extension. Then execute the SQL script using SQLPlus.

1. Create a user with username = *your_student_id* and password = cse4308 and grant necessary privileges to log in and execute DDL and DML statements. Then log in as that user.

2. Write an SQL statement to create a table '`STUDENT`' which has 4 attributes:

   - `ID` (Assign it as Primary Key)
   - `NAME` (Ensure that this is not `NULL`)
   - `DEPT_NAME` (Each department name must be three letters)
   - `TOT_CRED` (ensure that TOT_CRED is not greater than 180)

3. Write SQL statements to insert the following records into '`STUDENT`' table:

ID	NAME	DEPT_NAME	TOT_CRED
00128	Zhang	CSE	102
12345	Shankar	CSE	32
19991	Brandt	HIS	80
23121	Chavez	FIN	110
44553	Peltier	PHY	56
45678	Levy	PHY	46
54321	Williams	CSE	5
55739	Sanchez	MUS	38
70557	Snow	PHY	0
76543	Brown	CSE	58
76653	Aoi	EEE	60
98765	Bourikas	EEE	9
98988	Tanaka	BIO	120

4. Write SQL statements to perform the following queries:

   - Display all records of the '`STUDENT`' table.
   - Find names and departments of students who have completed 80 to 120 credits (inclusive).
   - Show the ID and name of students from CSE department.

- Find the name and credits of students from Physics (PHY) department for students who have completed less than 50 credits.
- Show the names of the departments which contain the students whose names are ‘Shankar’ or ‘Sanchez’.

5. The authority has decided to purge all of the students who have taken less than 10 credits. Delete those entries from the database.

6. Drop the ’`STUDENT`’ table with all its constraints.

7. **BONUS**: These tasks are not mandatory to complete and will add bonus marks if completed:

   - Show the name of the student with the highest total credit.
   - Find the sum of credit scores of students from Physics (PHY) department.