COMP 3311 Database Management Systems Spring 2015

Lab 4. Simple DDLs and DMLs, and enforcing constraints

Objectives of the Lab

- After this lab you should be able to
  - Issue simple Data Definition Language commands,
  - Issue simple Data Manipulation Language commands,
  - Know to apply simple integrity constraints.

Getting the lab SQL script file

- Download the lab4.sql file as follows
  - 1. login to an arbitrary machine csl2wkxx.cse.ust.hk where xx=01-40
  - 2. at the command prompt type

```
csl2wk01:lamngok:121> wget \
? http://course.cse.ust.hk/comp3311/labs/lab4.sql
```

- Log into Oracle database server using SQL*Plus with your password.

Running the lab SQL script

- Execute the script lab4.sql at the prompt

```
SQL> @lab4.sql
```

- The tables created last time were dropped.
- Some new tables are created.

Simple Data Definition Language 1

- The Data Definition language (DDL) is a language for specifying the database schemes (in the form of tables). It enables operations to be made on creating tables and altering the tables.
- You will learn the following DDLs in this lab.
  - CREATE
  - RENAME
  - DROP
  - ALTER

Simple Data Definition Language 2

- Creating a new table

  CREATE TABLE table_name (column1 datatype, column2 datatype, …);

```
create table department_facility
(  department_id        varchar2(4) not null,
    name  varchar2(40),
    no_of_projectors number(4),
    no_of_computers  number(5));
```

- Renaming an existing table

  RENAME old_table TO new_table

```
RENAME department_facility to test;
```

Simple Data Definition Language 3

- Dropping an existing table

  Drop TABLE table_name;

  DROP TABLE test;

- Adding new columns to an existing table

  ALTER TABLE table_name ADD (column1 datatype, column2 datatype,…);

  ALTER TABLE facility ADD (funding number(10));

Simple Data Definition Language 4

- Changing the data type of the column

  ALTER TABLE table_name MODIFY (column1 datatype, column2 datatype,…);

  ALTER TABLE facility MODIFY (funding varchar2(10));

- Deleting a column from an existing table

  ALTER TABLE table_name DROP (column1,column2,…);

  ALTER TABLE facility DROP (funding);

- We will cover how to enforce Integrity Constraints (IC) to the tables/columns by the DDL in the next lab.

Simple Data Manipulation Language 1

- The Data Manipulation language (DML) is a language for manipulating data in a database.
- You will learn the following DML in this lab.
  - INSERT
  - DELETE
  - UPDATE

Simple Data Manipulation Language 2

- Insert records to an existing table

  INSERT INTO table_name (column1, column2,…) VALUES (value1, value2,…)

  INSERT INTO facility (department_id, name, no_of_projectors, no_of_computers) VALUES ('COMP', 'Computer Science', 5, 150);

- You can omit the column names, if you are inserting records with all the columns present.

  INSERT INTO facility VALUES ('COMP', 'Computer Science', 5, 150);

Simple Data Manipulation Language 3

- By stating explicitly the columns, you can insert partial records with some of the columns being absent, as long as these columns do not have the “NOT NULL” constraint (will cover the “NOT NULL” constraint in details in the next lab).

```
INSERT INTO facility (department_id)
VALUES ('test');
```

Simple Data Manipulation Language 4

- Removing a record from an existing table

  DELETE FROM table_name [WHERE conditions]

  ```
  DELETE FROM facility WHERE department_id='test';
  ```

- The following statement removes all records from the table facility

  ```
  DELETE FROM facility;
  ```

Simple Data Manipulation Language 5

- Updating tuples of an existing table

```
UPDATE table_name SET column= value
[WHERE conditions];
```

```
UPDATE facility SET no_of_computers=200
WHERE department_id= 'COMP';
```

Enforcing Integrity Constraints 1

- We need to ensure that changes made to the database do not disrupt data consistency.
- One of the methods is to enforce integrity constraints on the database.

Enforcing Integrity Constraints 2

- Integrity constraints can be declared at the column level or at the table level.
- Column level constraints apply to the columns only. Each constraint involves one column.
- Table level constraints apply to the whole table. Usually involved multi-columns.
- In Oracle, column level constraints are placed right after the column definitions. Table level constraints are placed after all the definitions of the columns.

Enforcing Integrity Constraints 3

- The Constraint commands
  - PRIMARY KEY: specifies the column(s) that are used to uniquely indentify the rows(records) in a table.
  - FOREIGN KEY: specifies the column(s) that is/are being "borrowed" from another table and must be present in that table.
  - UNIQUE: indicates the column has unique values.
  - NOT NULL: indicates the column must have a value.
  - CHECK: place conditions (in the form of a predicate) on the column.

- Oracle Database allows applying the above constraints at the column level or at the table level (except for the NOT NULL constraint which can only be applied as a column level constraint).

Enforcing Integrity Constraints 4

- Examples:

```
CREATE TABLE staff (
id number(10) PRIMARY KEY,
age number(3) CHECK
                   (age between 0 and 65),
salary number(10)  CHECK
                   (salary>0));

CREATE TABLE work (
id number(10) REFERENCES staff(id),
firm_name VARCHAR2(100) NOT NULL,
Primary Key(id, firm_name));
```

Column Level Constraints

Table Level Constraint

Enforcing Integrity Constraints 5

- Examples:

The following two statements are identical. Note that all the constraints in the second CREATE statement were given names (in italic format).

```
CREATE TABLE work (
 id number(10) REFERENCES staff (id),
 firm_name VARCHAR2(100) NOT NULL,
Primary Key (id, firm_name)
 );

CREATE TABLE work(
id number(10),
firm_name VARCHAR2(100)
CONSTRAINT not_null NOT NULL,
CONSTRAINT f_key FOREIGN KEY (id) REFERENCES staff (id),
CONSTRAINT p_key Primary Key(id, firm_name)
);
```

Enforcing Integrity Constraints 6

- One can add or modify constraints in an existing table by their names, using the ALTER TABLE statement.

```
ALTER TABLE staff ADD CONSTRAINT test
CHECK (age between 20 and 40);

ALTER TABLE work MODIFY (firm_name
null);
```

Enforcing Integrity Constraints 7

- We can drop a non-primary key constraint

```
ALTER TABLE staff DROP CONSTRAINT test;
```

- We can also drop a primary key

```
ALTER TABLE work DROP PRIMARY KEY;
```

or add it back

```
ALTER TABLE work  MODIFY (PRIMARY KEY (id,firm_name));
```

- We need to remember the constraint name in order to drop it. The following query returns all the declared constraints.

```
SELECT constraint_name FROM user_constraints;
```

Conclusion

- We covered the following topics in this lab:
  - Simple DDLs and DMLs,
  - How enforcing integrity constraints.