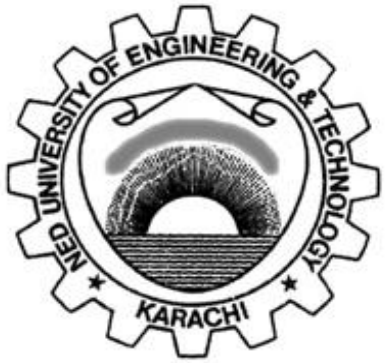
Kabeer Ahmed
SE-19028



Database Management System
(DBMS – 204)

Experiment # 09

CONSTRAINTS

Student Name: Kabeer Ahmed

Roll Number: SE-19028

Maximum Marks	Performance = 05	Viva = 05	Total = 10
Marks Obtained			
Remarks (if any)			

Experiment evaluated by

Instructor Name: Engr. Adiba Jafar

Signature and Date:

Kabeer Ahmed
SE-19028

Outcome

After completing this lesson, you should be able to do the following:

1. Describe constraints
2. Create and maintain constraints

What Are Constraints?

1. Constraints enforce rules at the table level.
2. Constraints prevent the deletion of a table if there are dependencies.

Constraints are used to specify rules for the data in a table. **Constraints** are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the table. If there is any violation between the **constraint** and the data action, the action is aborted.

The following constraint types are valid:

1. NOT NULL
2. UNIQUE
3. PRIMARY KEY
4. FOREIGN KEY
5. CHECK

Defining Constraints

CREATE TABLE [schema.]table(column datatype [DEFAULTexpr][column_constraint],...[table_constraint][,...]);
CREATE TABLE employees(employee_id NUMBER(6),first_name VARCHAR2(20), job_id VARCHAR2(10) NOT NULL,CONSTRAINT employees_emp_id_pk PRIMARY KEY (EMPLOYEE_ID));

Defining Constraints

1. Column constraint level:
   Column [CONSTRAINT constraint_name]constraint_type ,
2. Table constraint level:
   Column,[CONSTRAINT constraint_name] constraint_type(column, ...),

The NOT NULL Constraint

1. Ensures that null values are not permitted for the NOT NULL Constraint
2. The NOT NULL constraint ensures that the column contains no null values. Columns without the NOT NULL constraint can contain null values by default.

- CREATE TABLE deptb(
  deptno NUMBER(2) primary key,dname varchar2(13) NOT NULL,loc varchar2(14));
- CREATE TABLE empb(empno NUMBER(6) PRIMARY KEY,ename VARCHAR2(25) NOT NULL,job varchar2(9),mgr number(4), hiredate DATE,sal NUMBER(7,2),comm NUMBER(7,2),deptno number(2),foreign key(deptno)references deptb(deptno));

The UNIQUE Constraint

A **unique constraint** is an integrity **constraint** that ensures the data stored in a column, or a group of columns, is **unique** among the rows in a table. Typically, you apply the **unique constraints** to columns when you create the table using the inline **constraint** syntax as follows: CREATE TABLE table_name (...

Kabeer Ahmed
SE-19028

It is defined at either the table level or the column level.

CREATE TABLE EMP(employee_id NUMBER(6),last_name VARCHAR2(25) NOT NULL, salary NUMBER(8,2), commission_pct NUMBER(2,2), hire_date DATE NOT NULL,email VARCHAR2(25),
CONSTRAINT emp_email_uk UNIQUE(email));

The PRIMARY KEY Constraint

The PRIMARY KEY constraint uniquely identifies each record in a table. Primary keys must contain UNIQUE values, and cannot contain NULL values. A table can have only ONE primary key; and in the table, this primary key can consist of single or multiple columns (fields).

Is defined at either the table level or the column level
CREATE TABLE departments(department_id NUMBER(4),
department_name VARCHAR2(30),
manager_id NUMBER(6),
location_id NUMBER(4),
CONSTRAINT dept_id_pk PRIMARY KEY(department_id));

The FOREIGN KEY Constraint

A foreign key (FK) is a column or combination of columns that is used to establish and enforce a link between the data in two tables to control the data that can be stored in the foreign key table.

The FOREIGN KEY Constraint

Is defined at either the table level or the column level
CREATE TABLE empc(
empno NUMBER(4) primary key,
ename VARCHAR2(10) NOT NULL,job varchar2(9),mgr number(4),hiredate DATE NOT NULL,sal NUMBER(7,2),
comm NUMBER(7,2),deptno number(2)
CONSTRAINT emp_deptno_fk FOREIGN KEY (deptno)
REFERENCES deptb(deptno);

FOREIGN KEY Constraint Keywords

•FOREIGN KEY: Defines the column in the child table at the table constraint level
•REFERENCES: Identifies the table and column in the parent table
•ON DELETE CASCADE: Deletes the dependent rows in the child table when a row in the parent table is deleted
•ON DELETE SET NULL: Converts dependent foreign key values to null

The CHECK Constraint

A **check constraint** is a type of integrity **constraint** in SQL which specifies a requirement that must be met by each row in a database table. The **constraint** must be a predicate. It can refer to a single column, or multiple columns of the table.
•Defines a condition that each row must satisfy
•The following expressions are not allowed:

Kabeer Ahmed
SE-19028
–References to CURRVAL,NEXTVAL,LEVEL, and ROWNUM pseudo columns
–Calls to SYSDATE ,UID,USER, and USERENV functions
–Queries that refer to other values in other rows..., salary NUMBER(2)
CONSTRAINT emp_salary_min CHECK (salary > 0),...

Adding a Constraint Syntax

Use the ALTER TABLE statement to:

- Add or drop a constraint, but not modify its structure
- Enable or disable constraints
- Add a NOT NULL constraint by using the
  MODIFY clause

```
ALTER TABLE table
ADD [CONSTRAINT  constraint]type(column);
```

Adding a Constraint

Add a FOREIGN KEY constraint to the EMPLOYEES table to indicate that a manager must already exist as a valid employee in the EMPLOYEES table.

1. ALTER TABLE deptB ADD CONSTRAINT DEPTB_deptno_pk primary key(deptno);

2. ALTER TABLE empB ADD CONSTRAINT empb_deptno_fk FOREIGN KEY(deptno) REFERENCES deptB(deptno);

Table altered.

Dropping a Constraint

• Remove the DEPTNO constraint from the EMPLOYEES table.

```
ALTER TABLE DEPTB drop constraint deptb_deptno_pk;
ALTER TABLE     empB DROP CONSTRAINT  empb_deptno_fk;
```

Table altered.

• Remove the PRIMARY KEY constraint on the DEPARTMENTS table and drop the associated FOREIGN KEY constraint on the EMPLOYEES.DEPARTMENT_ID column.

```
ALTER TABLE DEPTB DROP PRIMARY KEY CASCADE;
```

Table altered.

Disabling Constraints

• Execute the DISABLE clause of the ALTER TABLE statement to deactivate an integrity constraint.
• Apply the CASCADE option to disable dependent integrity constraints.

```
ALTER TABLE EMPB DISABLE CONSTRAINT empB_deptno_pk CASCADE;
```

Table altered.

Enabling Constraints

•Activate an integrity constraint currently disabled in the table definition by using the ENABLE clause.

```
ALTER TABLE empB
ENABLE CONSTRAINT empB_deptno_pk;
```

Table altered.

•A UNIQUE or PRIMARY KEY index is automatically created if you enable a UNIQUE key or PRIMARY KEY constraint.

Cascading Constraints

•The CASCADE CONSTRAINTS clause is used along with the DROP COLUMN clause.

Kabeer Ahmed
SE-19028

• The CASCADE CONSTRAINTS clause drops all referential integrity constraints that refer to the primary and unique keys defined on the dropped columns.
• The CASCADE CONSTRAINTS clause also drops all multicolumn constraints defined on the dropped columns.

Cascading Constraints

Example

ALTER TABLE test1 DROP (pk) CASCADE CONSTRAINTS;
Table altered.
ALTER TABLE test1 DROP (pk, fk, col1) CASCADE CONSTRAINTS;
Table altered.

Viewing Constraints

Query the USER_CONSTRAINTS
table to view all constraint definitions and names.

```
SELECT constraint_name, constraint_type, search_condition FROM user_constraints
WHERE table_name = 'EMPLOYEES';
```

Viewing Constraints

Viewing the Columns Associated with Constraints

View the columns associated with the constraint names in the USER_CONS_COLUMNS view.

```
SELECT constraint_name, column_name
FROM user_cons_columns
WHERE table_name = 'EMPLOYEES';
Select * from user_cons_columns where  table_name=’EMPLOYEES’;
```

Kabeer Ahmed
SE-19028

LAB # 09
Constraints

1. Add a table-level PRIMARY KEY constraint to the EMP table on the ID column. The constraint should be named at creation. Name the constraint my_emp_id_pk.

```
SQL> ALTER TABLE EMP
  2  ADD CONSTRAINT my_emp_id_pk PRIMARY KEY (EMPNO);
ADD CONSTRAINT my_emp_id_pk PRIMARY KEY (EMPNO)
               *
ERROR at line 2:
ORA-02437: cannot validate (SYSTEM.MY_EMP_ID_PK) - primary key violated
```

2. Create a PRIMARY KEY constraint to the DEPT table using the ID column. The constraint should be named at creation. Name the constraint my_deptid_pk .
Hint:
The constraint is enabled as soon as the ALTER TABLE command executes successfully.

```
SQL> ALTER TABLE DEPT
  2  ADD CONSTRAINT my_deptid_pk PRIMARY KEY (DEPTNO);

Table altered.
```

3. Add a column DEPT_ID to the EMP table. Add a foreign key reference on the EMP table that ensures that the employee is not assigned to a nonexistent department. Name the constraint my_emp_dept_id_fk.

```
SQL> ALTER TABLE EMP
  2  ADD (DEPT_ID NUMBER(7));

Table altered.

SQL> ALTER TABLE EMP
  2  ADD CONSTRAINT my_emp_dept_id_fk
  3  FOREIGN KEY (DEPT_ID) REFERENCES DEPT(DEPTNO);

Table altered.
```

Kabeer Ahmed
SE-19028

4. Confirm that the constraints were added by querying the USER_CONSTRAINTS view. Note the types and names of the constraints. Save your statement text in a file called lab9_4.sql.

```
SQL> save e://lab9_4.sql;
Created file e://lab9_4.sql
SQL> get e://lab9_4.sql;
  1  SELECT constraint_name, constraint_type
  2  FROM user_constraints
  3* WHERE table_name IN ('EMP','DEPT')
SQL> @ e://lab9_4.sql;

CONSTRAINT_NAME
--------------------------------------------------------------------------------
C
-
SYS_C007466
C

MY_DEPTID_PK
P

SYS_C007467
C


CONSTRAINT_NAME
--------------------------------------------------------------------------------
C
-
MY_EMP_DEPT_ID_FK
R


SQL>
```

Kabeer Ahmed
SE-19028

5. Display the object names and types from the USER_OBJECTS data dictionary view for the EMP and DEPT tables. Notice that the new tables and a new index were created.

```
SQL> SELECT object_name, object_type
  2  FROM user_objects
  3  WHERE object_name LIKE 'EMP%'
  4  OR object_name LIKE 'DEPT%';

OBJECT_NAME
--------------------------------------------------------------------------------
OBJECT_TYPE
-----------------------
DEPT
TABLE

DEPT40
TABLE

EMP
TABLE


OBJECT_NAME
--------------------------------------------------------------------------------
OBJECT_TYPE
-----------------------
EMP3
TABLE
```

6. Modify the EMP table. Add a COMMISSION column of NUMBER data type, precision 2, scale 2. Add a constraint to the commission column that ensures that a commission value is greater than zero.

```
SQL> ALTER TABLE EMP
  2  ADD commision NUMBER(2,2)
  3  CONSTRAINT my_emp_comm_ck CHECK (commision>=0);

Table altered.
```