# DAY 11

```
PRN : 200243020003
```

## Creating and managing tables and adding constraints

#### 1. Create table copy_emp same as employees table

```
CREATE TABLE copy_emp AS
SELECT
    *
FROM
    EMPLOYEE;
```

#### 2. Populate the DEPT table with data from the DEPARTMENTS table. Include only columns that you need

```
CREATE TABLE dept AS
SELECT
    DEPARTMENT_ID,
    DEPARTMENT_NAME
FROM
    DEPARTMENTS;
```

#### 3. Create table EMP and add all necessary constraints.Include only the EMPLOYEE_ID, FIRST_NAME, LAST_NAME, SALARY, and DEPARTMENT_ID columns. Name the columns in your new table ID, FIRST_NAME, LAST_NAME, SALARY , and DEPT_ID, respectively.

```
CREATE TABLE emp (
    EMPLOYEE_ID NUMBER CONSTRAINT e_id PRIMARY KEY,
    FIRST_NAME VARCHAR2 (30),
    LAST_NAME VARCHAR2 (30),
    SALARY NUMBER,
    DEPARTMENT_ID NUMBER
```

#### 4. Truncate table EMP

```
TRUNCATE TABLE EMP;
```

### 5. Drop the EMP table

```
DROP TABLE EMP;
```

### 6. Now again create table EMP.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.

```
CREATE TABLE emp (
    EMPLOYEE_ID NUMBER,
    FIRST_NAME VARCHAR2 (30),
    LAST_NAME VARCHAR2 (30),
    SALARY NUMBER,
    DEPARTMENT_ID NUMBER,
    CONSTRAINT my_emp_id_pk PRIMARY KEY (EMPLOYEE_ID)
);
```

### 7. 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.

```
CREATE TABLE dept (
    id NUMBER CONSTRAINT my_deptid_pk PRIMARY KEY
);
```

### 8. Modify the EMP table to allow for longer employee last names. Confirm your mod ification.

```
ALTER TABLE emp MODIFY (last_name VARCHAR2 (50));

DESCRIBE emp;
```

### 9. Confirm that both the DEPT and EMP tables are stored in the data dictionary. (Hint: USER_TABLES)

```
SELECT
    table_name
FROM
    user_tables
WHERE
    table_name IN(DEPT, EMP);
```

### 10. Drop the FIRST_NAME column from the EMP table. Confirm your modification by checking the description of the table.

```
ALTER TABLE emp DROP COLUMN FIRST_NAME;

DESCRIBE emp;
```

**11. 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 depar tment. Name the constraint my_emp_dept_id_fk.**

```
ALTER TABLE emp
    ADD (dept_id NUMBER (7));

ALTER TABLE emp
    ADD CONSTRAINT my_emp_dept_id_fk FOREIGN KEY (dept_id) REFERENCES dept
(id);
```

**12. Confirm that the constraints were added by querying the USER_CONSTRAINTS view. Note the types and names of the constraints.**

```
SELECT
    constraint_name,
    constraint_type
FROM
    user_constraints
WHERE
    table_name IN(EMP, DEPT);
```

**13. 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**

```
SELECT
    object_name,
    object_type
FROM
    user_objects
WHERE
    object_name LIKE EMP %
    OR object_name LIKE DEPT %;
```

**14. 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 co mmission value is greater than zero**

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