CSE 311L(Database Management System)

LAB-Week 01(Part A)

Objectives:

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

- Create database tables
- Describe the data types that can be used when specifying column definition
- Table naming rules & Fields Datatypes

Table Naming Rules

Table names and column names:

- Must begin with a letter or underscore.
- Can be up to **64 characters**.
- Must contain only A–Z, a–z, 0–9, _, $, and #
- Must not duplicate the name of another object owned by the same user
- **Avoid reserved words** like SELECT, ORDER, etc.

Commonly used Data Types:

Data Type	Description
VARCHAR(n)	Variable-length string with a max length of n characters (up to 65,535 bytes). Efficient for strings with varying length.
CHAR(n)	Fixed-length string. Pads with spaces if shorter than n. Good for uniform-length data.
INT	Whole number (−2,147,483,648 to 2,147,483,647). Most common integer type.
DECIMAL(p,s)	Fixed-point number with precision p and scale s

FLOAT(p,d)	Approximate floating-point number. p is precision; d is decimals.
DOUBLE	Larger floating-point numbers than FLOAT
DATE	Date in YYYY-MM-DD format.
DATETIME	Date and time in YYYY-MM-DD HH:MM:SS
TIMESTAMP	Like DATETIME but time zone-aware and auto-updated (optional).
BINARY(n)	Fixed-length binary data.
BLOB	Binary data up to 65,535 bytes.

CREATE TABLE Statement

CREATE TABLE

Specifies a new base relation by giving it a name, and specifying each of its attributes and their data types

Syntax of CREATE Command:

CREATE TABLE <*table name***>** (<Attribute *A*1> <Data Type *D*1> [< Constarints>], <Attribute *A*2> <Data Type *D*2> [< Constarints>],
…….
<Attribute *A*n> <Data Type *D*n> [< Constarints>],
[<integrity-constraint1>, <integrity-constraint k>]);

- A constraint NOT NULL may be specified on an attribute

Example: CREATE TABLE DEPARTMENT (
DNAME VARCHAR(10) NOT NULL,
DNUMBER INT NOT NULL,
MGRSSN CHAR(9),
MGRSTARTDATE CHAR(9));

Integrity Constraints Description

An integrity constraint is a declarative method of defining a rule for a column of a table.

Constraint	Description
NOT NULL	Ensures that a column **cannot contain NULL** values.
UNIQUE	Ensures that **all values in a column are different** (no duplicates). (but can contain null values)
PRIMARY KEY	Uniquely identifies each record in a table. Combines NOT NULL + UNIQUE
CHECK	Ensures that values in a column satisfy a **specific condition**.
DEFAULT	Sets a **default value** for a column when no value is provided.
AUTO_INCREMENT	Automatically generates a unique number (usually for IDs).

Referential Integrity Constraints

Different tables in a relational database can be related by common columns, and the rules that govern the relationship of the columns must be maintained. Referential integrity rules guarantee that these relationships are preserved.
The following terms are associated with referential integrity constraints.

Term	Definition
Foreign key	Ensures **referential integrity** by linking two tables
Referenced key	The unique key or primary key of the same or different table that is referenced by a foreign key.

Dependent or child table	The table that includes the foreign key. Therefore, it is the table that is dependent on the values present in the referenced unique or primary key
Referenced or parent table	The table that is referenced by the child table's foreign key. It is this table's referenced key that determines whether specific inserts or updates are allowed in the child table

Specifying the unique, primary key attributes, secondary keys, and referential integrity constraints (foreign keys).

```
Ex: CREATE TABLE DEPT (
DNAME VARCHAR(10) NOT
NULL, DNUMBER INTEGER
NOT NULL, MGRSSN
CHAR(9), MGRSTARTDATE
CHAR(9), PRIMARY KEY
(DNUMBER), UNIQUE
(DNAME),
FOREIGN KEY (MGRSSN) REFERENCES EMP(SSN));
```

We can specify RESTRICT, CASCADE, SET NULL or SET DEFAULT on referential integrity constraints (foreign keys)

```
Ex: CREATE TABLE DEPT (
DNAME VARCHAR(10) NOT
NULL, DNUMBER INTEGER
NOT NULL,
MGRSSN CHAR(9), MGRSTARTDATE
CHAR(9), PRIMARY KEY (DNUMBER),
UNIQUE (DNAME),
FOREIGN KEY (MGRSSN) REFERENCES EMP
    ON DELETE SET DEFAULT ON UPDATE CASCADE);
```

Activity 01:

Write SQL statement for create the 'Emps' table: (company2. Schema)
(You will be given demo of the code. Just follow it and ask questions, if there is any)

Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(6)
FIRST_NAME		VARCHAR2(20)
LAST_NAME	NOT NULL	VARCHAR2(25)
EMAIL	NOT NULL	VARCHAR2(25)
PHONE_NUMBER		VARCHAR2(20)
HIRE_DATE	NOT NULL	DATE
JOB_ID	NOT NULL	VARCHAR2(10)
SALARY		NUMBER(8,2)
COMMISSION_PCT		NUMBER(2,2)
MANAGER_ID		NUMBER(6)
DEPARTMENT_ID		NUMBER(4)

Syntax Column_Constraint: [constraint name]{[NOT] NULL| {UNIQUE | PRIMARY KEY}|
REFERENCES table [(column)]| CHECK (condition)}

Ex:SQL>create table subjects(subdesc varchar(20) constraint const_name NOT NULL..);
Ex2:SQL>create table subjects(subdesc varchar(10) constraint subject_pk PRIMARY KEY);

Syntax Table_Constraint: [constraint name]{{UNIUQE | PRIMARY KEY} (**Col, [, col]…)|**

FOREIGN KEY **(Col, [, Col}…)** REFERENCES **table [(Col {, Col]…)|** CHECK (condition)}

Ex1: SQL>Create table studm**arks (…,constraint studmarks_chk** Check(stdate<=enddate));

Ex2: syntax for table constraint

SQL> Create table studmars (sno number (5), constraint studmarks_fk FOREIGN KEY (sno) references students (sno));

DROP TABLE

Used to remove a relation (base table) and its definition.

The relation can no longer be used in queries, updates, or any other commands since its description no longer exists

Example: DROP TABLE DEPENDENT;

ALTER TABLE:

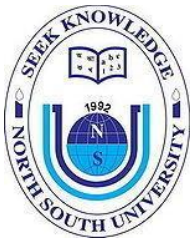
Used to add an attribute to/from one of the base relations drop constraint -- The new attribute will have NULLs in all the tuples of the relation right after the command is executed; hence, the NOT NULL constraint is *not allowed* for such an attribute.

Example: ALTER TABLE EMPLOYEE ADD JOB VARCHAR(12);

The database users must still enter a value for the new attribute JOB for each EMPLOYEE tuple. This can be done using the UPDATE command.

DROP A COLUMN (AN ATTRIBUTE)

ALTER TABLE COMPANY.EMPLOYEE DROP ADDRESS CASCADE; All constraints and views that reference the column are dropped automatically, along with the column. ALTER TABLE COMPANY.EMPLOYEE DROP ADDRESS RESTRICT; Successful if no views or constraints reference the column. ALTER TABLE COMPANY.DEPARTMENT ALTER MGRSSN DROP DEFAULT;



CSE 311L(Database Management System)

LAB-Week 01(Part B)

Objectives:

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

- Insert rows into the created table
- Create Department Table
- Execute a basic SELECT statement

The `INSERT` Statement Syntax

```
INSERT INTO table [(column [, column...])]
VALUES (value [, value...]);
```

Activity 01:

Write SQL statement for create the 'Depts' table:

Name	Null?	Type
DEPARTMENT_ID	NOT NULL	NUMBER(4)
DEPARTMENT_NAME	NOT NULL	VARCHAR2(30)
MANAGER_ID		NUMBER(6)
LOCATION_ID		NUMBER(4)

Activity 02:

Write SQL statement for INSERT two employees' data into the employees table you create earlier.

Activity 03:

Write SQL statement for INSERT two Departments' data into the Departments table you just created.