| C:\Users\faisal\Pictures\NSU_pic_download\n91267046457_2661.jpg |
| --- |

Objectives:



**Table Naming Rules**

Table names and column names:

* Must begin with a letter
* Must be 1–30 characters long
* Must contain only A–Z, a–z, 0–9, \_, $, and #
* Must not duplicate the name of another object
* owned by the same user
* Must not be an Oracle server reserved word

**Data Types:**

|  | **Data type** | **Description** |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | | |
|  | VARCHAR2(LEN) | Variable length char. Values up to width length. Maximum width | | |
|  |  | is 4000 chars. |  |  |
|  |  |  | | |
|  | CHAR(LEN) | Fixed length char. Values of width length. Default width is 1. | | |
|  |  | Maximum length is 200. |  |  |
|  |  |  | | |
|  | NUMBER | Floating point with precision of 38 significant digits. | | |
|  |  |  | | |
|  | NUMBER (Precision, | Precision represents Maximum significant digits allowed, which | | |
|  | Scale) | may not exceed 38. Scale is the number of decimal places on the | | |
|  |  | right of the decimal point. |  |  |
|  |  |  | | |
|  | DATE | Date values in the range 1-1-4712 B.C to 31-12-4712 AD. | | |
|  |  |  |  | |
|  | LONG | Variable length char. Values up to 2 GB. | Only one LONG col. is | |
|  |  | allowed per table. You cannot use LONG datatype in functions, | | |
|  |  | WHERE clause and sub queries. |  |  |
|  |  |  | | |
|  | RAW AND LONG RAW | Equivalent to VARCHAR2 and LONG respectively, used for storing | | |
|  |  | digital sound or graphics images. |  |  |
|  |  |  |  | |
|  | CLOB, BLOB, NCLOB | Used to store large char. And binary | Objects .Each can | |
|  |  | accommodate upto 4 GB. |  |  |
|  |  |  |  |  |
|  | BFILE | Stores a pointer to an external file. |  |  |
|  |  |  | | |
|  | TIMESTAMP | It is an extension of Date datatype. It stores year, month, day, | | |
|  |  | hour, minute and second values |  |  |
|  |  |  |  |  |
|  |  |  |  | |

**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 A constraint NOT NULL may be specified on an attribute

Example: CREATE TABLE DEPARTMENT (

DNAME VARCHAR(10) NOT NULL,

DNUMBER INTEGER 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. Oracle supports the following integrity constraints:

* NOTNULL constraints for the rules associated with nulls in a column
* UNIQUE key constraints for the rule associated with unique column values
* PRIMARYKEY constraints for the rule associated with primary identification values
* FOREIGNKEY constraints for the rules associated with referential integrity. Oracle supports the use of FOREIGNKEY integrity constraints to define the referential integrity actions, including:
  + - Update & delete no action
    - Delete CASCADE
    - Delete SET NULL
* CHECK constraints for complex integrity rules

**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 | The column or set of columns included in the definition of the referential integrity constraint that reference a referenced key |
| 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),

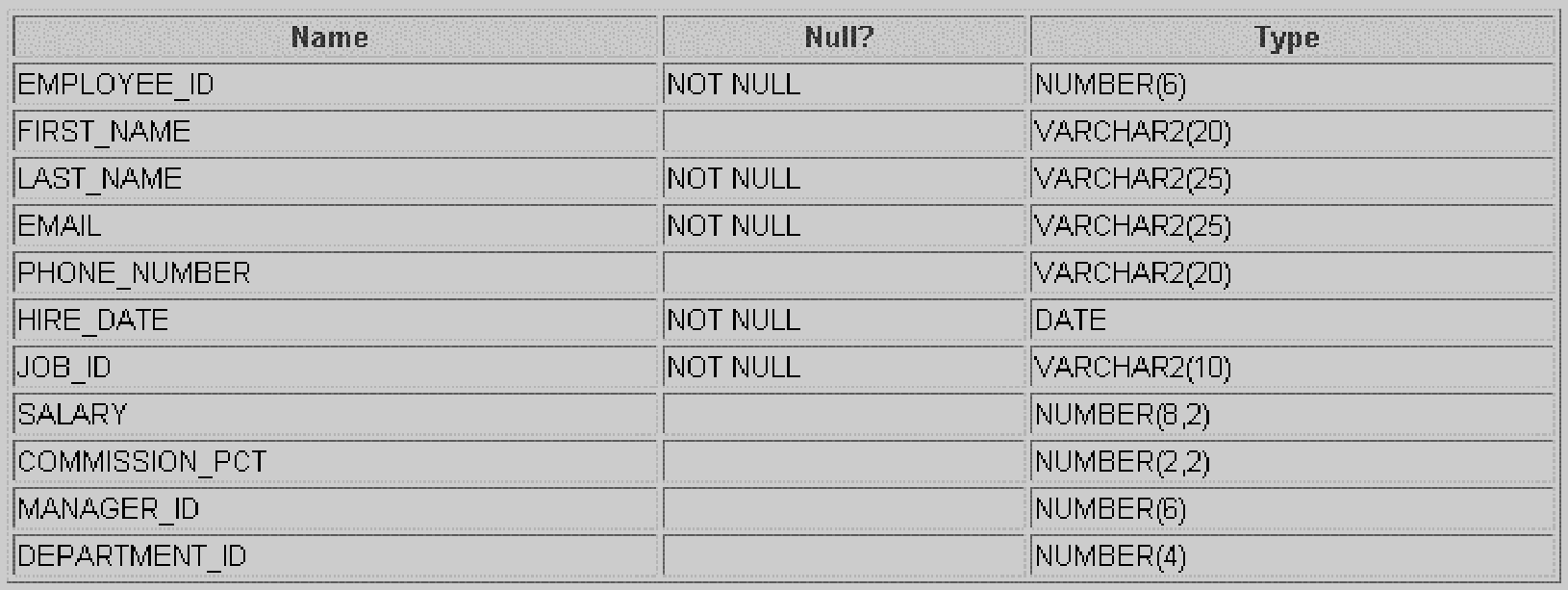
FOREIGN KEY (MGRSSN) REFERENCES EMP

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)



**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 VARCHAR2 (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.

| C:\Users\faisal\Pictures\NSU_pic_download\n91267046457_2661.jpg |
| --- |

Objectives:



**The INSERT Statement Syntax**

**INSERT INTO *table* [(*column* [*, column...*])]**

**VALUES *(value* [*, value...*]);**

Activity 01:

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

.

INSERT

INTO Emps VALUES(100, 'Steven','King', 'SKING','515.123.4567', '2006-06-17', 'AD\_PRESS',24000, NULL, NULL, 90),

(101, 'Neena','Kochar', 'NKOCHAR','515.123.4568', '2008-09-21', 'AD\_VP',17000, NULL, 100, 90),

(102, 'Lex','De Haan', 'DEHAAN','515.123.4569', '2009-01-13', 'AD\_VP',17000, NULL, 100, 90),

(103, 'Alexander','Hunold', 'AHUNOLD','590.423.4567', '2008-01-03', 'IT\_PROG',9000,NULL, 102, 60),

(104, 'Bruce','Ernst', 'BERNST','590.423.4568', '2009-05-21', 'IT\_PROG',6000,NULL, 103, 60),

(107, 'Diana','Lorentz', 'DLORENTZ','590.423.5567', '2008-02-17', 'IT\_PROG',4200,NULL, 103, 60),

(124, 'Kevin','Mourgos', 'KMORGOS','650.123.5234', '2012-11-16', 'ST\_MAN',5800,NULL, 100, 50),

(141, 'Treena','Rajs', 'RRAJS','650.121.5234', '2004-10-17', 'ST\_CLERK',3500,NULL, 124, 50),

(142, 'Curtis','Davies', 'CDAVIES','121.123.5234', '2007-01-29', 'ST\_CLERK',3100,NULL, 124, 50),

(143, 'Randall','Matos', 'RMATOS','121.123.5234', '2008-03-15', 'ST\_CLERK',2600,NULL, 124, 50),

(144, 'Peter','Vargas', 'PVARGAS','121.123.5234', '2008-07-09', 'ST\_CLERK',2500,NULL, 124, 50),

(149, 'Eleni','Zlotkey', 'EZLOTKEY','44.1344.429018', '2014-01-29', 'SA\_MAN',10500,.2, 100, 80),

(174, 'Ellen','Abel', 'EABEL','44.1644.429017', '2004-05-11', 'SA\_REP',11000,.3, 149, 80),

(176, 'Jnathon','Taylor', 'JTAILOR','44.1644.429021', '2008-03-24', 'SA\_MAN',8600,.2, 149, 80),

(178, 'Kimberely','Grant', 'KGRANT','44.1644.429023', '2009-05-24', 'SA\_MAN',7000,.15, 149, NULL),

(200, 'Jennifer','Whalem', 'JWHALEN','515.123.4444', '2003-09-17', 'ADD\_ASST',4400,NULL, 101, 10),

(201, 'Michael','Hartstein', 'MHARSTEIN','515.123.5555', '2008-02-17', 'MK\_MAN',13000,NULL, 100, 20),

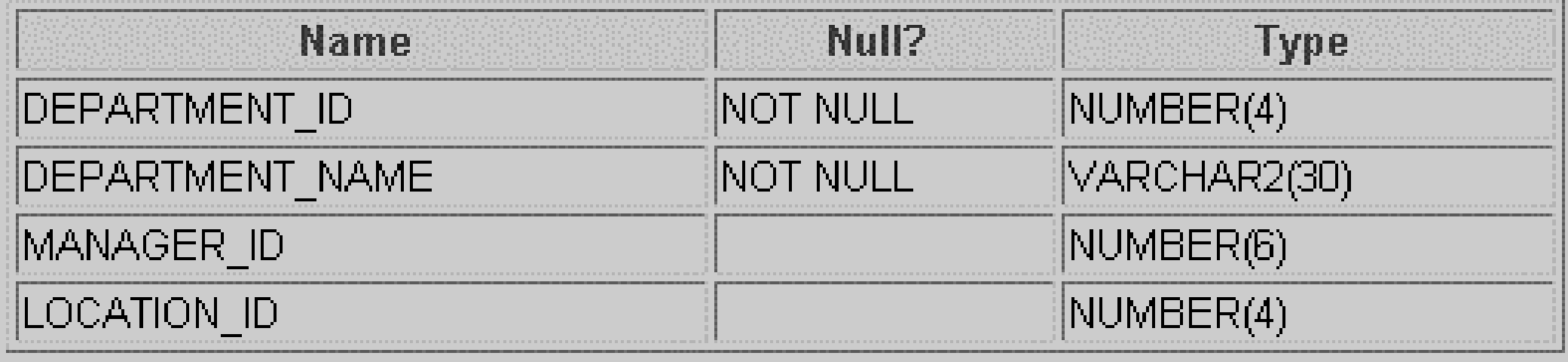
(202, 'Pat','Fay','PFAY','603.123.6666', '2010-08-17', 'MK\_REP',6000,NULL, 201, 20),

(205, 'Shelley','Higgins', 'SHIGGINS','515.123.8050', '2007-06-07', 'AC\_MGR',12000,NULL, 101, 110),

(206, 'William','Gietz', 'WGIETZ','515.123.8181', '2007-07-07', 'AC\_ACCOUNT',8300,NULL, 205, 110)

Activity 02:

Write SQL statement for create the 'Depts' table:



Activity 03:

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

