19/09/23

→ Oracle - RDBMS (III)
SQL → Table (Relations)

Relations:
Unary — $R = \{(2,3)(3,4)(4,5)(5,6)\}$
Binary — $R: A \to A \quad A = \{1,2,3,4,5,6\}$
Ternary

DBMS → Relation (Table)
→ Student (RollNo, name, class, CGPA, address, mphone)

Table → Columns → Attributes
→ Rows → Records

→ attributes describe a record data

	RNo	Name	Class	CGPA	add	mphone
1						
2						
3						
4						
5						

→ Storing the required data acc. to the task given.

* → Faculty (FID, FName, Department, DOJ, cadre)

→ To design a database; we must identify

SQL is case sensitive
Wrt Data

Commands are not case-sensitive

the entities required for task.

* Commands in SQL:

→ Data Definition Language (DDL)
↳ Create & modify structure

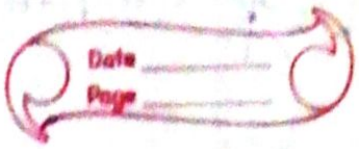
* Create → create the table
* Alter → change the structure
(datatype modification, add any missed or delete data)
* Truncate → delete the data (complete) but structure is left
* Drop → delete data as well as table.
* Describe → describes the attributes and its datatypes

→ Data Manipulation Language (DML)
↳ modify / update the data

* Insert → insert data into relation
* Update → change the existing data
* Delete → delete the data
↳ We can delete the specified data or records

** Select → access the data from database, table/tables using any specific conditions
70 - 80% of queries are written using select

Privileges → Read Privilege
Write Privilege



↳ Data Control language (DCL)

* ADD → add the constraints
* REVOKE → take the privilege
* Commit → if we close the application; we will lock the database

* Create Table without constraints:

Syntax:

```
create table <tablename>
        (<column1name><datatype>(<size>),
        <column2name><datatype>(<size>),
        :
        <columnnname><datatype>(<size>));
```

* Datatypes:

fixed length datatype

char : characters, strings - (1-255 characters)
Ex: name char[25].

ansi standard, oracle standard

variable length datatype

varchar/varchar2 : - 1-255 characters
↳ length can be ↑sed / ↓sed
Ex: name varchar2[20] → we can use certain for one set & other for others

** varchar is 50% slower than char

Number (p,s)
- p → precision
- s → no. of decimal points

sal number(10,2)
- total we can take 10 digits
- 8 for before decimal &
- 2 after decimal

Numeric(p,s) → absolute zero value

Date → date attributes
Date(DD-MON-YY)
Ex: Date(23-JAN-81)
- for display we can use various functions

* Create a Student table:

*write in single line

```
create table
Student (Rollno varchar2(15),
         Name char(30),
         Class varchar2(10),
         CGPA Numeric(3,2),
         Branch char(10),
         Address varchar2(50),
         PhoneNo Number(10,0));
```

Table Created

table name - alphabet & digits
- upto 30 characters
- underscore can be used
- 1st alphabet is letters
- cannot use reserved words
- no duplication of table names

Real Numbers → precision & decimal points compulsory

Integer → not required

(2)
```
create table
    Faculty (FID varchar2(10),
             FName char(30),
             Branch char(10),
             DOJ Date(
             cadre char(20));
```
table Created.

(3) create table
```
    Course (CID varchar2(10);
            CName char(20),
            Credits Number);
```
table created.

(4) Create table
```
    Department (DNO Number,
                Dname char(20),
                Location char(20));
```
table created.

Syntax: desc <tablename>

DDL table <tablename>

Syntax: insert into <tablename> (Rno, Name, Branch) (column names)
(attributes which are not specified are NULL)
values(' ', ' ',);

varchar2 (used)
char

→ for all values:
insert into <tablename> values(, ,);
1 row created.
select * from tablename;
set lines 1000;
select Rno, Name from tablename;

24/03/2023

→ Create tables with constraints:

* create table Account
(CID varchar2(10) primary key,
Name char(20) check(name = upper(Name))
Balance Numeric(10,2)
check(balance>0),
check(CID like("C%")));

String fns:
upper(); like("C%") → start with C
like("%C") → end with C

like ("_c%") → 2nd character is c.

→ at column level; give only one constraint

but at table level; we can give any no. of constraints.

→ on delete cascade: If particular record in primary table is deleted; this command deletes the particular foreign key record from secondary tables.

→ on delete SET NULL: If a value in primary key column is made NULL; then its value in secondary tables is also set to NULL.

* create table dept1 (DNo varchar2(10) primary key Name char(10), loc char(10) unique);

* create table Student (RNO varchar2(10) primary key class varchar2(10) default '2/4 CSE-A', CGPA Numeric(4,2), DNo varchar2(10) // references dept1 on delete cascade);

table level.

if we do not specify the attribute; by default it takes primary key; else if we want to give any other column; then that attribute shd be unique.

create table course
(CID varchar2(10) primary key
CName char(20), credits numeric(3),
RNo varchar2(10),
// foreign key (RNo references
Student on delete set
null));

Column level

* DML:

→ update: modifies the data in the table.

update <tablename> set <columnname>
= <value> where [<cond>]
↓
optional

→ update Student set CGPA = 9.5;
update Student set CGPA
where RNO = '025';

update Student set CGPA = 9.5,
name = 'ABC' where
RNO = '025';

To use DML commands; we must describe the table so as to know the datatypes of the attributes.

* delete: delete a record, specified records or all the records

```
delete from dept1 where dno='30';
delete from dept1;
```

→ We either delete entire record and we can't delete only few values in a record.

* Select:

→ display data

→ We can display data from multiple tables with specific conditions

```
select <columnnames> from <table name>
                [where <condition>]
```