

## Lab 1

### Basic Schema Definition – Data Definition Language (DDL)

#### Basic Datatypes:

Integer or int	4 bytes	typical choice for integer	-2147483648 to +2147483647
Numeric (precision,scale)	variable	user-specified precision, exact	<p>up to 131072 digits before the decimal point; up to 16383 digits after the decimal point</p> <p>The <b>precision</b> of a Numeric is the total count of significant digits in the whole number, that is, the number of digits to both sides of the decimal point. So, the number 23.5141 has a precision of 6 and a scale of 4. Integers can be considered to have a scale of zero.</p> <p>The <b>scale</b> of a <b>Numeric</b> is the count of decimal digits in the fractional part, to the right of the decimal point.</p>
varchar (n)			variable-length with limit
char (n)			fixed-length, blank padded

#### 1. Creating your first table (without any integrity constraints)

```
Create table student
( ID          varchar(5),
  Name        varchar(15)
  Cgpa        numeric(3,1))
```

**1. Create the tables using the DDL statements given below: with integrity statements)**

```
create table department
(dept_name    varchar (20),
 building     varchar (15),
 budget       numeric (12,2),
 primary key (dept_name));
```

```
create table course
(course_id    varchar (7),
 title       varchar (50),
 dept_name   varchar (20),
 credits     numeric (2,0),
 primary key (course_id),
 foreign key (dept_name) references department);
```

```
create table instructor
(ID          varchar (5),
 name        varchar (20) not null,
 dept_name   varchar (20),
 salary      numeric (8,2),
 primary key (ID),
 foreign key (dept_name) references department);
```

**2. Remove table** – removes the table as well as data present in it.

Drop table *table\_name*;  
e.g. Drop table department;

**Data Manipulation Language (DML) : Basic**

**3. Inserting records (data) in the table :**

**First form** : Here we need to specify values for all the columns and in the same order as they were specified at the time of table creation. For a particular column that can take null values we can specify its value as NULL also.

Insert data into the tables as follows:

insert into instructor values ('22222','Einstein','Physics',95000)

Or

**Second Form** : In the second form we can use column names in the SQL statement and this gives us a lot of flexibility. Firstly, we can omit any column and secondly, we don't need to maintain any order.

insert into instructor (id, name, dept\_name) values ('22222','Einstein','Physics')

This is to insert only specific data. Here salary field is not entered.

Or

**Third Form** : This is to insert multiple records in one insert statement.

insert into instructor(id,name,dept\_name,salary) values

('22222','Einstein','Physics',95000), ('12121','Wu','Finance',90000);

ID	name	dept_name	salary
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

(a) The *instructor* table

dept_name	building	budget
Comp. Sci.	Taylor	100000
Biology	Watson	90000
Elec. Eng.	Taylor	85000
Music	Packard	80000
Finance	Painter	120000
History	Painter	50000
Physics	Watson	70000

(b) The *department* table

**Figure 1.2** A sample relational database.

course_id	title	dept_name	credits
BIO-101	Intro. to Biology	Biology	4
BIO-301	Genetics	Biology	4
BIO-399	Computational Biology	Biology	3
CS-101	Intro. to Computer Science	Comp. Sci.	4
CS-190	Game Design	Comp. Sci.	4
CS-315	Robotics	Comp. Sci.	3
CS-319	Image Processing	Comp. Sci.	3
CS-347	Database System Concepts	Comp. Sci.	3
EE-181	Intro. to Digital Systems	Elec. Eng.	3
FIN-201	Investment Banking	Finance	3
HIS-351	World History	History	3
MU-199	Music Video Production	Music	3
PHY-101	Physical Principles	Physics	4

**Figure 2.2** The *course* relation.

## 4. Modification of the data

### Deletion:

```
delete from instructor;  
delete from instructor where dept_name = 'Finance';  
delete from instructor where salary between 1300 and 2000;
```

### Updation:

```
update instructor set salary = salary + 1000 where salary < 7000;  
update instructor set salary = 1000 where name = 'Raj';
```

## Queries on Single Relations

- 1) Retrieve the contents of the instructors table.  
select \* from instructors;
- 2) Find the name of all the instructors  
select name from instructors;
- 3) Find the department name of all the instructors  
select dept\_name from instructors;
- 4) Select clause with arithmetic expression  
select name, salary\*1.5 from instructors;
- 5) Where clause with predicates  
select name from instructor where dept\_name='Comp.Sci.' and salary>7000;

### Self-try

1. Retrieve the contents of the department table.
2. List the various courses offered by the Computer Science department.
3. List the various courses offered by the Computer Science department with 3 credits.
4. Change the credits for a particular course.
5. Delete from courses those courses with less than 3 credits.
6. Create a table student with attributes Rollnumber of type varchar(10), Name of type varchar(25), State of type varchar(10)
7. Insert suitable data to student table.
8. Add a column CGPA to student table and insert suitable data.

## **Example Alter statements in PostgreSQL**

To add a column to a table:

1. ALTER TABLE student ADD COLUMN tuition\_fee numeric(9,2);
2. ALTER TABLE student ADD COLUMN bus\_fee numeric(9,2);

To drop a column from a table:

3. ALTER TABLE student DROP COLUMN bus\_fee;

To change the types of two existing columns in one operation:

4. ALTER TABLE student  
    ALTER COLUMN state TYPE varchar(80),  
    ALTER COLUMN name TYPE varchar(30);

### **To rename a column**

5. alter table student rename column state to state\_1;

### **To rename a table**

6. alter table student rename to student\_123;