

# RDBMS Concepts

What is database?

- Any collection of related information
  - ↳ Phone Book
  - ↳ Shopping list
  - ↳ Todo list
  - ↳ Your 5 best friend
  - ↳ fb's user base
- DB can be stored in different ways
  - ↳ on paper
  - ↳ in your mind
  - ↳ on pc
  - ↳ powerpoint
  - ↳ sql
- Collection of related information that can be stored in different ways

Computers + Database = ❤

## Amazon.com

- ↳ keeps track of products, reviews, purchase order etc.
- ↳ info is extremely valuable
- ↳ security is essential
  - ↳ card (credit), Address, phone
- ↳ info stored on pc

## Shopping list

- ↳ keeps track of consumer products that need to be purchased
- ↳ convenience sake
- ↳ not important
  - ↳ piece of paper, memory (brain)

Computers are great at keeping track of large amounts of information

April 2017

## Database Management System (DBMS)

March						
W	S	M	T	W	T	F
10	5	6	7	8	9	10
11	12	13	14	15	16	17
12	19	20	21	22	23	24
13	26	27	28	29	30	31

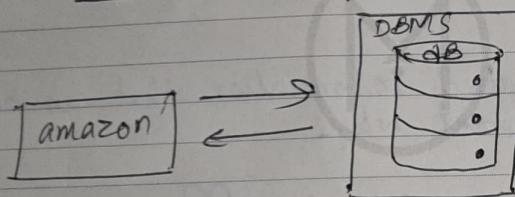
Ram Navami (India)

4 Tuesday  
(094 - 271) Wk 14

↳ Special s/w prof. that helps users create & maintain a DB.

- ↳ Makes it easy to manage large amount of info.
- ↳ Handles Security
- ↳ Backups
- ↳ Importing / exporting data
- ↳ Concurrency
- ↳ Interacts with s/w applications

### Amazon.com DB diagram



amazon.com will interact w. the DBMS in order to create, read, update & delete info.

## RDBMS

- Earlier data was stored in bank

↳ data was completely handwritten & has lots of books which collectively held this data.

↓ After computer era

file base system

↳ Still some problem with data redundancy.

April						
W	S	M	T	W	T	F
15/18/30				1	2	3
14	2	3	4	5	6	7
15	9	10	11	12	13	14
16	16	17	18	19	20	21
17	23	24	25	26	27	28

May						
W	S	M	T	W	T	F
18				1	2	3
19	7	8	9	10	11	12
20	14	15	16	17	18	19
21	21	22	23	24	25	26
22	28	29	30	31		

↳ Data related to banking sector was stored in multiple locations which cost data to be redundant across diff locations.

↳ Also, possibility was that only certain amount of files were edited & the remaining files were left as such.

Problems:- Which was the right file containing right information?

↳ Security?

⇒ file base - data storage sys. was humongous task to maintain.

Soln :- DBMS

↳ All the data that was stored w.r.t. to a particular business was stored on one centralized repository as your db.

↳ And since this is the only DB sys. which stores the entire data, we could enforce security based policies on this particular data.

↳ Only authorized users can access this data.

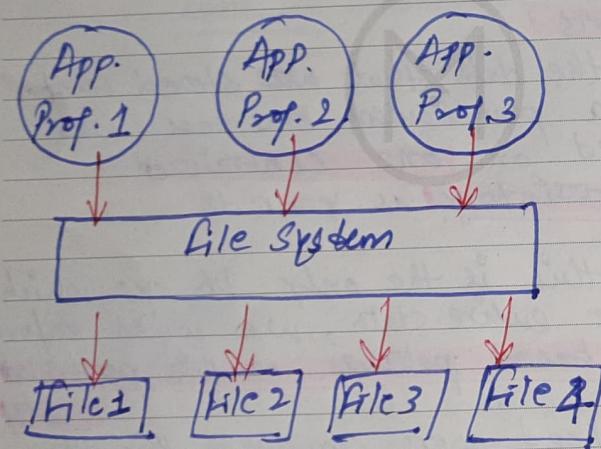
Information System :- Organized collection of HW, SW, supplies, policies, procedures & people, which stores process & provides access to information.

April 2017

data in plain text formate

## Traditional Approach - File based System

- 6 Thursday (096 - 269) Wk 14
- ↳ info is stored in flat files, which are maintained by the file system under the operating systems control.
  - ↳ App. programs go through the file sys. in order to access these flat files.
  - ↳ Records consist of various fields, which are delimited by a space, comma, pipe, or any special character etc.



- ↳ We are able to free up all that space by moving all the data on the computer.
- ↳ No need to search for data in filing room.

- ↳ Still diff. to produce report across sales, product & customer data, becz they are maintained on a separate file sys.

Traditional Approach

when necessary

W	S	M	T	W	T	F	S
13/18/30				1	2	3	4
14	2	3	4	5	6	7	8
15	9	10	11	12	13	14	15
16	16	17	18	19	20	21	22
17	23	24	25	26	27	28	29

↳ In older days, app(s) were developed in an ad-hoc manner.

↳ Data files are developed for individual applications

↳ In Traditional approach, app. profs are file dependent. on the files on which the data is stored.

↳ Any change in data req. change in app.  
↳ like file format

↳ Date Redundancy & date Isolation

↳ data can be duplicated into 2 or more files.

↳ all the related data are scattered in various files having diff. file format & hence writing new application become diff. in retrieving data.

DBMS - Intro: Problem of traditional approach has been overcome by DBMS.

DB :- collection of logically related data (with a description of this data), designed to meet the needs of an organization.

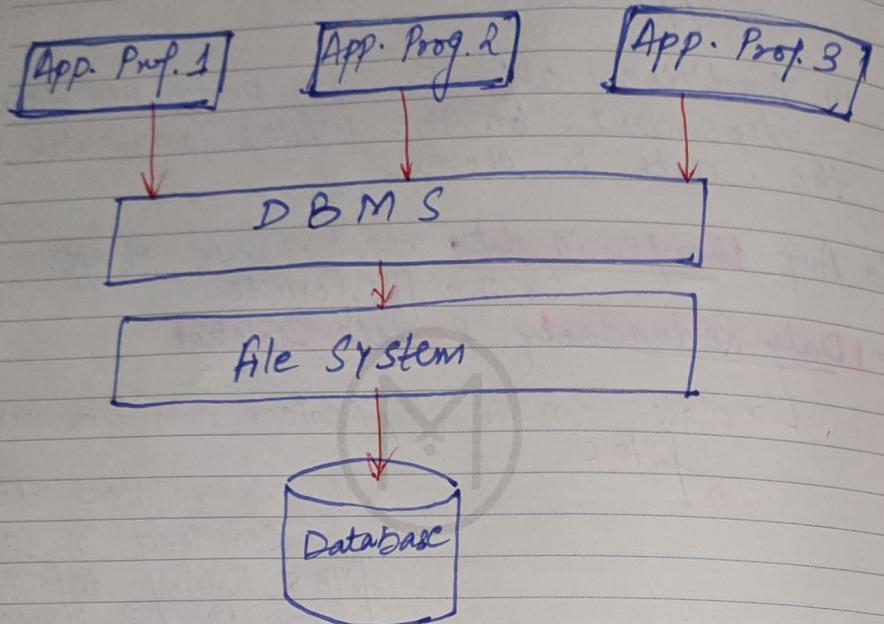
W	S	M	T	W	T	F	S
18	1	2	3	4	5	6	7
19	8	9	10	11	12	13	14
20	14	15	16	17	18	19	20
21	21	22	23	24	25	26	27
22	28	29	30	31			

May  
18 19 20 21 22 23 24 25 26 27  
(097 - 268) Wk 14

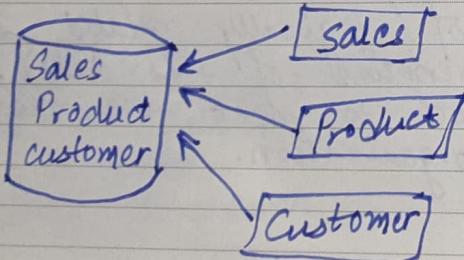
April 2017

DBMS → SW that enable users to define, create & maintain the DB & provides controlled access to the DB.

8 Saturday  
0598 - 267 Wk 14



- ↳ We don't need to store data in all the machines
- ↳ easier to generate report across sales, products & customers, bcz all data is stored in centralized place.



## Advantages of DBMS

- ↳ Centralization of info. management
- ↳ Data shared by diff. group of users & app. programs.
- ↳ Representation b/w data of complex relationship.
- ↳ Integrity constraint handling.
- ↳ Advanced facilities for backup & recovery.

1. Sharing of data
2. Enforcement of security
3. Enforcement of development & maintenance standards.
4. Reduction of redundancy
5. Avoidance of inconsistency across files
6. Maintenance of integrity.
7. Data independence.

## Database Users

Application Programmers :- Developers who write app. progs to interact with DB.  
↳ can be written in C++, JAVA, Python

Sophisticated users :- who interact with the system by forming their requests in a DB query language.

Each such query is submitted to query processor whose function is to break down by invoking one of the DML statement → instructions that storage manager understood previously.

End user : interact w. the sys. by invoking one of the permanent app. prog that DB has been written previously.

April 2017

10

Monday

(100 - 265) Wk 15

**DBA :-** (DB administrator). Users who manage the DB like installation of DB, managing users & DB performance.

**Data Model :-** Way of explaining the logical layout of the data & the relationship of various parts to each other on the whole.

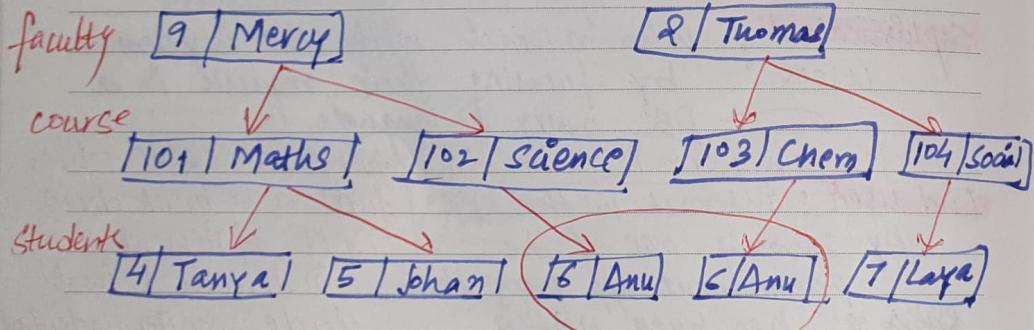
- Classification :-**
- Hierarchical
  - N/P
  - Relational

### Hierarchical Data Model

Data is represented by a tree structure

Can't handle many-many relations

Anomalies in Insert, Delete & update operations.



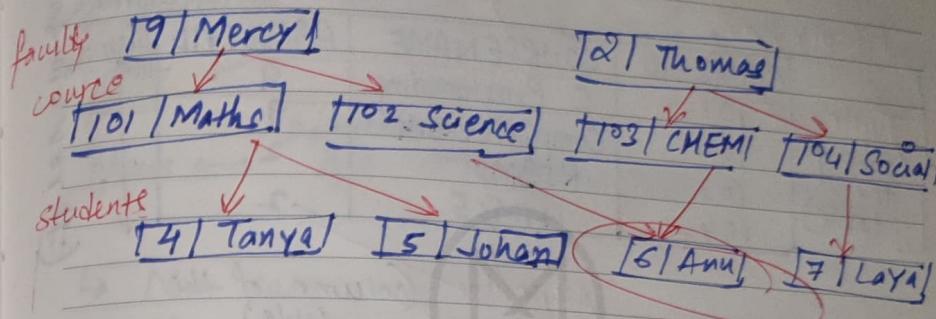
problem with insert, del op's

March						
S	M	T	W	T	F	S
10	5	6	7	8	9	10
11	12	13	14	15	16	17
12	19	20	21	22	23	24
13	26	27	28	29	30	31

April						
S	M	T	W	T	F	S
13/18/30	1	2	3	4	5	6
14	2	3	4	5	6	7
15	9	10	11	12	13	14
16	10	17	18	19	20	21
17	23	24	25	26	27	28

### N/W Data Model

Data is represented by records & pointers → Addresses many-many rel. → Insert, Delete, Update → Complex opt possible design



**Relational Data Model** → used widely  
→ stored in rows & columns  
→ no uses of pointers  
→ More flexible  
where the data is kept in tables → & easy to use  
foreign key  
→ almost any item of the data can be accessed more quickly than the other  
→ uses relational algebra concept.

Primary key

FACULTY ID	FACULTY NAME
9	Mercy
2	Thomas

Uniquely identified

COURSE ID	COURSE NAME
101	C Prog.
102	Comp. NW
103	RDBMS
104	HTML5

Pointers  
is a problem!

May						
S	M	T	W	T	F	S
18	1	2	3	4	5	6
19	7	8	9	10	11	12
20	14	15	16	17	18	19
21	21	22	23	24	25	26
22	28	29	30	31		

Tuesday

(101 - 264) Wk 15

April 2017

## Relationship b/w Tables.

12

Wednesday

(002 - 263) Wk 15

Link b/w tables can be update by using **Foreign key**, where the primary key of parent table becomes a column of child table.

March						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

April						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Sinhala & Tamil New Year (Aluth Avuruddha) / Vaisakhi (Sri Lanka / India)

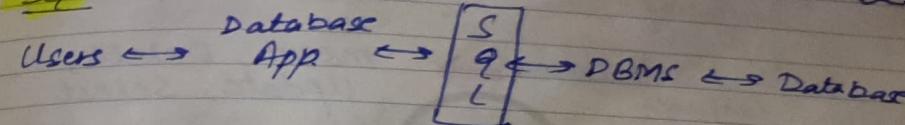
## Relational DBMS

May						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

- ↳ System S/W, that lets you create, update & administer a relational DB.
- ↳ uses Structured Query Language (SQL) to access the DB.

↳ RDBMS Products are Oracle, DB2, & MySQL etc

### SQL



- ↳ Not case sensitive
- ↳ SQL statements are used to retrieve & update data in a database
- ↳ SQL is an ANSI (American National Standards Institute) standard computer language for accessing & manipulating DB systems

## Standard & Best Practices of SQL

- Object Naming Conventions.
  - ↳ Use Pascal notation
  - ↳ Example: Products, Customers
- Column Names:
  - ↳ Use the singular form of nouns
  - ↳ FirstName, address
- Each table must have a primary key
- Use upper case for all SQL keywords
  - ↳ SELECT, INSERT, UPDATE, WHERE, AND, OR, LIKE etc.

## Relational Data Model - Terminology

Relation (Table)

Primary key			Domain X30			range of values that can be stored for attr. ut
COURSEID	COURSENAME	FACULTYID	Attribute (Column of the table)	Degree	# columns in the table	
101	C Programming	9				
102	Comp. Network	9				
103	RD8MD	2				
104	HTML5	2				

## Properties of Relations

1. No Duplicate Tuples
2. Tuples are unordered
3. Attributes are unordered
4. Attribute values are Atomic

April 2017

March						
S	M	T	W	T	F	S
9			1	2	3	4
10	5	6	7	8	9	10
11	12	13	14	15	16	17
12	13	14	15	16	17	18
13	14	15	16	17	18	19
14	15	16	17	18	19	20
15	16	17	18	19	20	21
16	17	18	19	20	21	22
17	18	19	20	21	22	23
18	19	20	21	22	23	24
19	20	21	22	23	24	25
20	21	22	23	24	25	26
21	22	23	24	25	26	27
22	23	24	25	26	27	28
23	24	25	26	27	28	29
24	25	26	27	28	29	30
25	26	27	28	29	30	31

14

Friday

(104 - 261) Wk 15

→ Do not use white space in identifiers  
Example: first name, last name

Good Friday

Bengali New Year / Mosadi (India)

→ Use parentheses to increase readability  
• WHERE (color = 'red') AND (size = 1 OR size = 2)

April						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
16	17	18	19	20	21	22
17	18	19	20	21	22	23
18	19	20	21	22	23	24
19	20	21	22	23	24	25
20	21	22	23	24	25	26
21	22	23	24	25	26	27
22	23	24	25	26	27	28
23	24	25	26	27	28	29
24	25	26	27	28	29	30
25	26	27	28	29	30	31

Vishu 2nd Day (India)

## Data Definition Language

### Creating & Managing Tables

Saturday 15

(105 - 260) Wk 15

defines the structure of DB

1. **CREATE** → To create new db objects.
2. **ALTER** → To modify an existing structure of the db objects.
3. **TRUNCATE** → To remove all data from db obj.
4. **DROP** → To remove the db obj. from db.

for further explanation ; Insurance Management System (IMS) is used.

Customer	Policy	Policy Enrollment
Id	PId	EnrollmentId
CName	PName	Id
PhoneNo	PeriodInYears	PId
Email	MinAmountPerMonth	Amount
Address		DueDate
		PaidDate
		Penalty

Data Types	Type of Data	
Varchar (size)	['John', 'Tom 123', 'Alex @#']	used to store variable length character data
Char (size)	['John', 'Tom 123', 'Alex @#']	fixed length character data
Number (p)	[123, 45678]	- integer
Number (p,s)	[789.453, 8.14]	integer & float
Data	['17-Oct-20']	Date

### Components of SQL

Data Definition Language → Create, Alter, Truncate, Drop

Data Manipulation Language → Insert, Delete, Update, Select

Transaction Control Language → Commit, Rollback

Date Control Language → Grant, Revoke

### Data Definition Language

↳ All DB objects like TABLE, VIEW, SEQUENCE, INDEX are created using 'CREATE' statement of DDL.

↳ Modification of object's structure is done using 'ALTER'

↳ Removing of data structure are done using 'TRUNCATE' and 'DROP' respectively.

DDL statements are AUTO COMMIT

April 2017

16

Sunday

(106 - 259) Wk 16

## Create a Table

- Syntax : CREATE TABLE tablename(  
    colname1 datatype, colname2,  
    datatype, .... colnameN  
    datatype);

Example :

```
CREATE TABLE customer (CId
    number(10), Cname varchar(25),
    phoneno number(10), Email
    varchar(25), address varchar(35))
```

## Naming Convention.

- Only letters, number & underscore are allowed in name.
- But we should start the name of the table with a meaningful identifier.

X   
 - Asdkgab  
 - Employee details } invalid  
 - 123

✓   
 - Employee - Details } valid  
 - Employee  
 - Employee - V1

March						
W	S	M	T	W	T	F
9			1	2	3	4
10	5	6	7	8	9	10
11	12	13	14	15	16	17
12	19	20	21	22	23	24
13	26	27	28	29	30	31

Easter Sunday (India)

April						
W	S	M	T	W	T	F
13/18/30			1	2	3	4
14	2	3	4	5	6	7
15	9	10	11	12	13	14
16	15	16	17	18	19	20
17	22	23	24	25	26	27

Family Day / Easter Monday (Kenya / South Africa)

May						
W	S	M	T	W	T	F
18			2	3	4	5
19	7	8	9	10	11	12
20	14	15	16	17	18	19
21	21	22	23	24	25	26
22	28	29	30	31		

## Record Insertion

I member I have Same no. Id	CID	CNAME	PHONENO	Email	Address
	1	TOM	9818	t@gmail.com	Chennai
	2	John	208	j@gmail.com	Delhi
	1	Ram	442	r@gmail.com	Pune

⇒ problem ⇒ when you want to identify record uniquely.  
 ↓

we can restrict it by having a constraint.

## Constraint

Rules enforced on data column of the tables.  
 These are used to limit the type of data that can go into the table.  
 This ensures the accuracy & reliability of the data in the DB.

## Types of constraints :-

1. Primary key
2. Foreign key
3. Unique
4. Not null
5. Check

A table can have one primary key, but it can have 'N' no. of foreign keys, unique, not null & check constraints.

2 columns joined together can be made as single primary key.

April 2017

18

Tuesday

(108 - 257) Wk 16

→ unique P not null

March						
W	S	M	T	W	T	F
9			1	2	3	4
10	5	6	7	8	9	10
11	12	13	14	15	16	17
12	19	20	21	22	23	24
13	26	27	28	29	30	31

April						
W	S	M	T	W	T	F
1				1	2	3
2	3	4	5	6	7	8
3	10	11	12	13	14	15
4	16	17	18	19	20	21
5	23	24	25	26	27	28
6	29	30	31			

CID	CNAME	PHONENO.	ADDRESS
1	Tom	9818	Chennai
2	John	208	Delhi
3	Ram	442	Pune

Here can we make CID as a primary key. ⇒ It will not allow any duplicate & NULL values

## UNIQUE

If we don't want an email id to be duplicated, can we mark as email id as primary key?

No!

Some customers might not have an email id. But we can mark it as unique

↓  
No duplicate but  
NULL ✓

## NOT NULL

Can we mark NAME as unique? No!  
(common Name)

NOT NULL ensure that there is no null but duplicate are allowed.

## CHECK

ensures the attribute has only a +ve value.

Can we ensure that whether the user is providing a +ve value for the amount to be paid?

## FOREIGN KEY

- for customer table, CID is the primary key
- for policy table, PID is the primary key
- for policy enrollment, ENROLMENTID is p.key

Customer	CID	CNAME	PHONENO	EMAIL	ADDRESS
	1	Tom	9876	t@xyz.com	Chennai
	2	John	8765	j@xyz.com	Delhi

Policy	PID	PNAME	PERIODINYEARS	MINAMOUNTPER MONTH
	NBP	Money Back Plan	20	1000
	PP	Personal Protect	15	1500

Policy Enroll.	ENROLMENT ID	CID	PID	DUEDATE	PAD DATE
	101	3	NBP	12-Dec-2017	11-Dec-2017
	102	1	PP	15-Mar-2018	13-Mar-2018

In policy enrollment table, there is an entry for CID as 3, but there are no customer as ID = 3.

This problem can be resolved using a foreign key

- There referred table is called the parent table
- The table with the foreign key is called child table.

May	S	M	T	W	T	F
18		1	2	3	4	5
19	7	8	9	10	11	12
20	14	15	16	17	18	19
21	21	22	23	24	25	26
22	28	29	30	31		

Wednesday	19
(109 - 256) Wk 16	

April 2017

20 Thursday

[110 - 255] Wk 16

March						
S	M	T	W	T	F	S
9		1	2	3	4	5
10	5	6	7	8	9	10
11	12	13	14	15	16	17
12	19	20	21	22	23	24
13	26	27	28	29	30	31

April						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
16	23	24	25	26	27	28
17	24	25	26	27	28	29

May						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
21	22	23	24	25	26	27
22	29	30	31			

## CONSTRAINT NAME

Friday

[110 - 254] Wk 16

- On creating a table with a constraint, oracle provides default name to it.
- The constraint names are stored in the built in table called "USER\_CONSTRAINTS".
- User can override the default constraint name with the user defined constraint name.
- Example:

```
CREATE TABLE Policy (Pid varchar2(10)
CONSTRAINT pk_pid primary key, Pname
varchar2(25), PPeriodInYears varchar2(25),
NOT NULL, MinAmount number(10)
CONSTRAINT ck_minamount check(MinAmount > 0))
```

ALTER → used to change the structure of the table.

```
CREATE TABLE PolicyEnrollment (EnrollmentId
number(5) primary key, Cid number(10)
REFERENCES customer(Cid), Pid varchar2(10),
DueDate date, PaidDate date, Penalty
number(10));
```

When we created a table PolicyEnrollment, we forgot to give policyId as a foreign key in policy enroll. table while creating table!

Alter table PolicyEnrollment add foreign key (Pid)
references Policy(Pid);

- The primary or unique column of the parent table can be create as the foreign key column in the child table.

We can mark CID & PID as foreign key to ensure integrity.

ENROLLMENTID	CID	PID	DUEDATE	PAIDDATE
101	1	MBP	12-Dec-2017	11-Dec-2017
102	2	PP	15-Mar-2018	13-Mar-2018

- Customer & policy are called parent table while policy enrollment table is called the child table.

## CREATE TABLE WITH CONSTRAINT

⇒ CREATE TABLE Customer (Cid number(10) PRIMARY KEY, CName varchar2(25) NOT NULL, phoneNo number(10), email varchar2(25) UNIQUE);

⇒ CREATE TABLE Policy (Pid varchar2(10) PRIMARY KEY, Pname varchar2(25), PPeriodInYears varchar2(25) NOT NULL, MinAmount number(10) CHECK (MinAmount > 0));

⇒ CREATE TABLE PolicyEnrollment (EnrollmentId number(5) PRIMARY KEY, Cid number(10) REFERENCES customer(Cid), Pid varchar2(10), DueDate date, PaidDate date, Penalty number(10));

April 2017

March						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

22 Saturday  
(112 - 253) wk 16

- Add a new column / constraint
- Remove an existing "

- Rename the existing column
- Increase or decrease the column size
- Change the column data type.

To add a column age to customer table

ALTER TABLE customer ADD age number;

To inc.↑ the column size of email to 30

ALTER TABLE customer MODIFY email varchar(30);

To change the column name from email to email id.

ALTER TABLE customer RENAME COLUMN email TO emailId;

## ALTER - SYNTAX

To change the datatype of a particular column

ALTER TABLE tablename MODIFY columnname newdatatype;

To remove a column from the table.

ALTER TABLE tablename DROP COLUMN column name;

To remove a constraint

ALTER TABLE tablename DROP CONSTRAINT constraintname;

April						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

To enable / disable a constraint

ALTER TABLE tablename ENABLE / DISABLE CONSTRAINT constraintname;

## TABLE LEVEL CONSTRAINT

Constraints can be column level or table level. Column level constraints are applied only to one column, whereas table level constraints are applied to the whole table.

## TRUNCATE

Removes all rows from the table

Restriction → You cannot truncate the table if it is linked with another table.

Syntax → TRUNCATE TABLE <table\_name>;

Example → TRUNCATE TABLE customer;

## DROP

Drops the entire table structure

Syntax → DROP TABLE <TABLE\_NAME>;

Example → DROP TABLE Customer

→ In truncate only the data is removed, whereas in drop the entire structure is removed.



Table



Table after Truncate

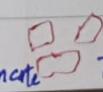


Table after drop

April 2017

March						
W	S	M	T	W	T	F
9				1	2	3
10	5	6	7	8	9	4
11	12	13	14	15	16	17
12	19	20	21	22	23	18
13	26	27	28	29	30	25

## Practise :-

### Creating Table

24 Monday

(114 - 251) Wk 17

Column Name	Datatype	Size	Constraint	Constraint name
User_id	Number	11	Primarykey	PK_USERS
Name	Varchar2	20		
Address	Varchar2	100		
Phno	Number	11		
Emailid	Varchar2	30		

Create table users(

```
user_id number(11),
name varchar2(20),
address varchar2(100),
phno number(11),
emailid varchar2(30),
constraint pk_users primary key(user_id)
);
```

## Alter Table

Write a query to add a new column, LiveTracker of datatype char(1). This will be used to track the live location of the bus, for the journey.

desc buses;

Name	Null ?	Type
BUS-NO	NOT NULL	NUMBER(11)
BUS-NAME		VARCHAR2(20)
TYPE		VARCHAR2(20)
TOTAL-SEAT		NUMBER(11)
RAIL-SEAT		NUMBER(11)