

to perform addition of 2 numbers

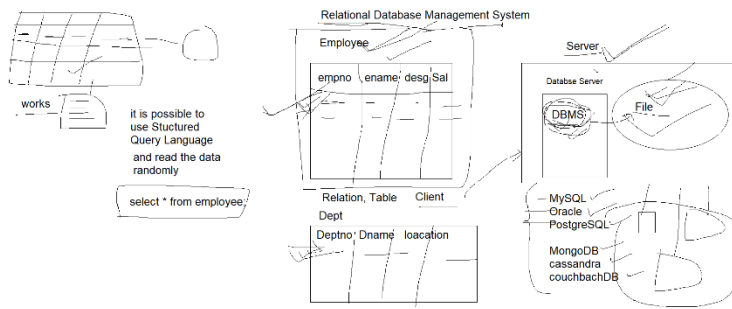
34+23=57

Memory

To store the data in persistent storage device, so that it will remain available all the time

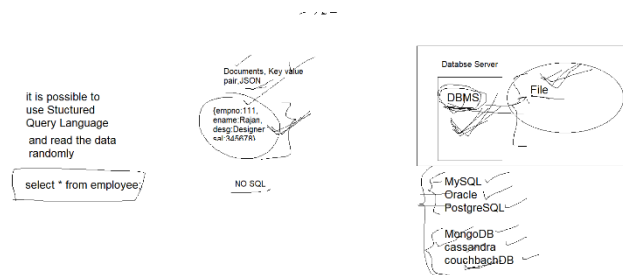
2 ways to store the data in persistent storage device

1. store it file in fix format file ---- flat
  - a. always read and write data in sequence, random access to data is not possible
  - b. to modify data, we need to read data in memory line by line modify it and overwrite the file or create new file
2. Databases
  - a. Types of data bases
    - i. RDBMS- Relational Database Management System
      1. oracle, MySQL, postgreSQL, SQL server
    - ii. NOSQL ---- to unstructured data
      1. MongoDB, CouchbaseDB, Cassandra
    - iii. GraphDB--- to get the display in graph format
      1. Neo4J
    - iv. MemDB ---- If we store the data in Memory
      1. VoltDB, MEM-DB
3. SQL ----- Structured Query Language
4. features RDBMS
  - a. Client server architecture
  - b. Secure ---- unless you have username and password you cannot access data
  - c. easy to use
  - d. avoids redundancy----- Duplicity of data
  - e. Managing data is easy
  - f. RDBMS application can be used from programs Java, Python, PERL, PHP
  - g. It is scalable--→ number of clients can be increased very easily and fast
  - h. It supports Transaction Control----- Most of financial application or where you need security and Transaction control then we use RDBMS  
e.g transfer fund ----→ money gets withdraw from source account before it gets transferred to destination a/c then network goes, or connection time out
  - i. Structured data ----- When data need to be in some specified format only then we call the data as Structured data



## 5. NOSQL

- data is stored in unstructured manner --- data can be entered in any format , like pictures or text or emoji's  
usually, it is used in social media application
- less secure- as compared to RDBMS
- and it is faster as compared to RDBMS
- Scalable



## 6. GraphDB

- Show the data in graph format
- if you want display, it as network

## 7. MemDB

- When the size of data is small
- If you need faster access to data
- Example ---- In Research projects

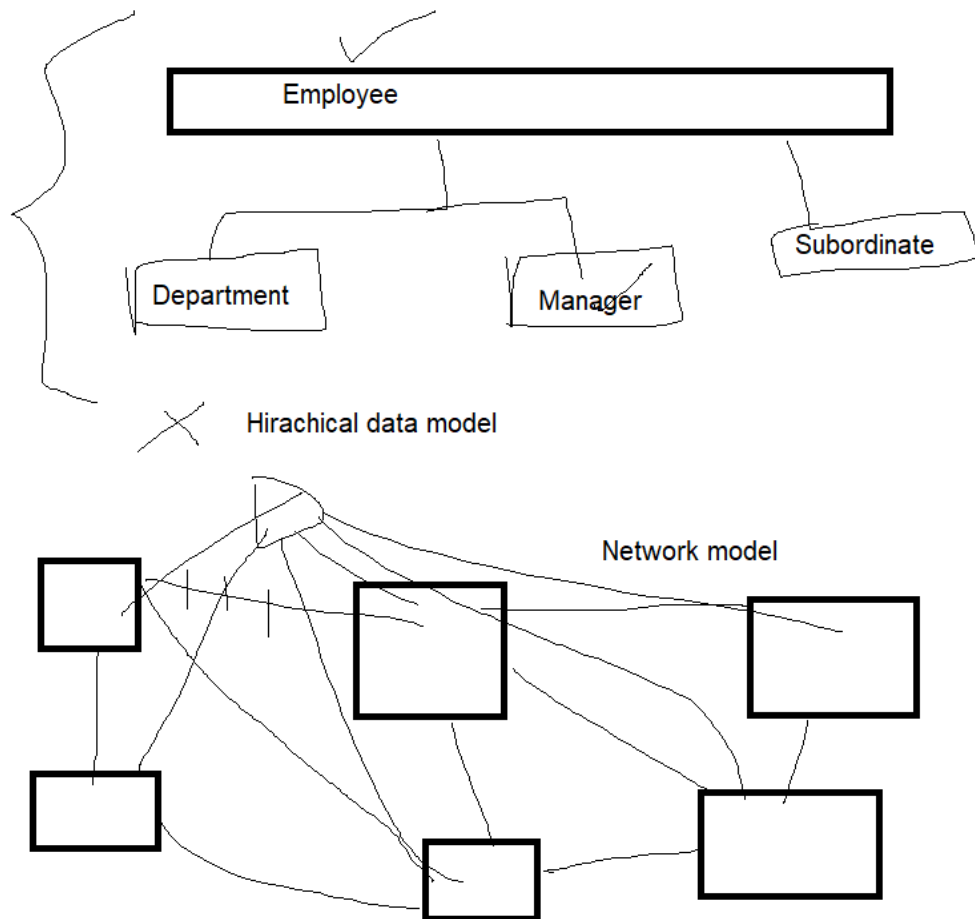
### DrawBacks of MySQL-----

1. MySQL does not support intersection operation, reference cursors, instead of trigger, statement level triggers
2. In MySQL very large databases are less efficient as compared to Oracle

### Structure of Data

To store the data the way it is organized is call as data model

- Hierarchical model
- Network model
- Relational Model



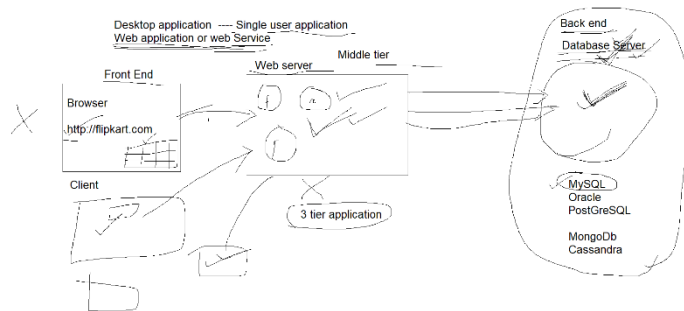
Employee Table or Employee Relation

Empno	Ename	Sal	deptno
1234	Rajas	222222	20
1111	Revati	333333	10

3333	Deven	5678	20

### Dept Relation

Deptno	Dname	Location
10	HR	Mumbai
20	Admin	Pune
30	Accounts	Pune



a/c no	Name	Mobile	Address	Email Address	Balance	Type of Account	BranchCode
1	Kishori	2222	Aundh	kk@gmail.com	100000	Saving	000101
2	Kishori	2222	Baner	kk@gmail.com	200000	current	000101
3	Kishori	2222	Baner	kk@gmail.com	250000	Demat	000101
4	Revati	333	Aundh	<a href="mailto:Rev@gmail.com">Rev@gmail.com</a>	230000	Saving	000101
5	Rajan	4444	Baner	<a href="mailto:Raj@gmail.com">Raj@gmail.com</a>	450000	Saving	000101

Redundancy ----- Repetition of data

Normalization Rules ----- helps to design data model

a/c no	Balance	Type of Account	BranchCode	Custid
1	100000	Saving	000101	1000
2	200000	current	000101	1000
3	250000	Demat	000101	1000
4	230000	Saving	000101	1001
5	450000	Saving	000101	1002

custid	Name	Mobile	Address	Email Address
1000	Kishori	2222	Aundh	kk@gmail.com
1001	Revati	333	Aundh	<a href="mailto:Rev@gmail.com">Rev@gmail.com</a>
1002	Rajan	4444	Baner	<a href="mailto:Raj@gmail.com">Raj@gmail.com</a>

### Why to use database

- Data is stored in centralized server, hence can be shared between applications
- data is stored in structured manner, in relation format, so retrieval of data is easy
- Correctness of data will be maintained
- redundancy is reduced so memory is efficiently uses
- Transaction control----- ACID property
- Secure

### What is ACID property

- Atomicity→ every transaction gets executed as single unit
- Consistency → After every transaction data is in correct state
- Isolation -→ any user when logs in should read same data
- Durability -→ longer period of time correctness or consistency of data will be maintained

1	Kishori	200000	
2	Rajan	300000	

---→ Kishori wants to transfer 50000 to Rajan

Atomicity -----→ to run these steps as a single unit of transaction

Read Source a/c balance and check whether sufficient for transfer

withdraw the amount

deposited it in Rajan's account

### Consistency

After completion of transaction balance in both the accounts should correct

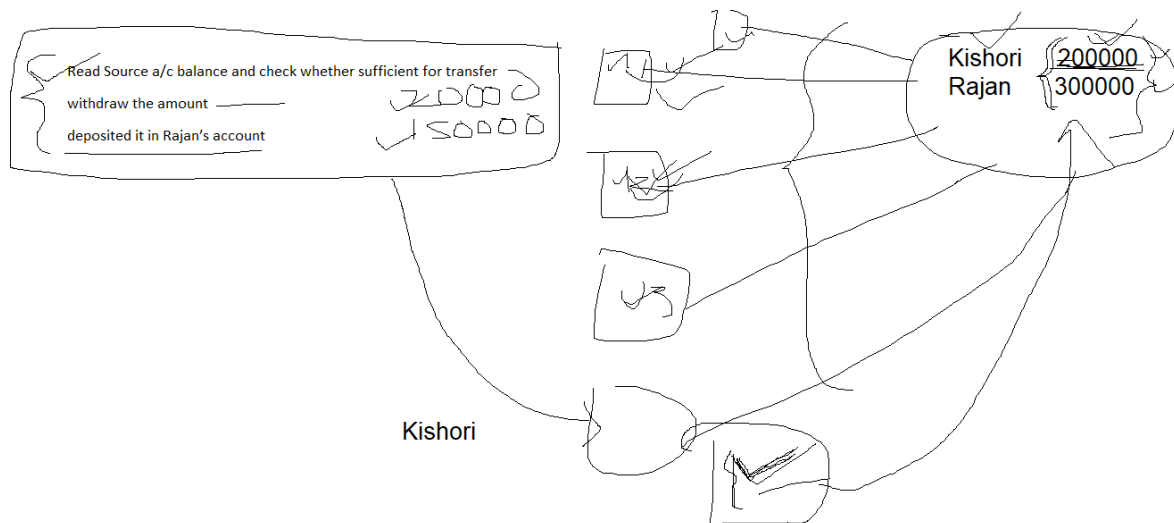
Kishori's balance ----- 150000

Rajan's balance -----350000

## Durability

---- valid service should be provided for longer of time

## Isolation



## Types of keys are in database

customer table

custid	Name	Mobile	Address	Email Address
1000	Kishori	2222	Aundh	kk@gmail.com
1001	Revati	333	Aundh	<a href="mailto:Rev@gmail.com">Rev@gmail.com</a>
1002	Rajan	4444	Baner	<a href="mailto:Raj@gmail.com">Raj@gmail.com</a>
1003	Kishori	2222	Aundh	kk@gmail.com

student

Studid	moduleid	Module name	Marks
1	100	JAVA	96
1	101	Database	97
1	102	Python	96
2	100	JAVA	98
2	101	Database	98

Employee

empid	Name	Mobile	Address	Email Address	Adhar card	Passport number	Pan card number
1000	Kishori	2222	Aundh	kk@gmail.com	222222	11111	Aa23456
1001	Revati	333	Aundh	<a href="mailto:Rev@gmail.com">Rev@gmail.com</a>	222223	null	Bb223423
1002	Rajan	4444	Baner	<a href="mailto:Raj@gmail.com">Raj@gmail.com</a>	222244	33333	null
1003	Kishori	2222	Aundh	kk@gmail.com	222111	44444	Dd342

1. primary key ----- minimal number of columns which identifies the row uniquely is primary key.

primary key for customer ----- custid

primary key for student ---- studid + module id

primary key for employee ---- empid

a. simple primary key --- if primary key contains one column

b. composite primary key ---- if primary key contains multiple columns

2. Alternate key-----all candidate keys which did not get selected as primary key is called as Alternate key

Alternate key for employee----→ passport number, adhar num, pan num

3. candidate key----- If there are more than one minimal subset which identifies the row uniquely, then all these are candidates for primary key

employee table---- empid, passport number, adhar num, pan num

4. Super key ---→ any combination of columns which identifies the row uniquely is called as super key

customer table----custid+name, custid, custid+address, custid+name+address, custid+email ,.....

student table ----- studid+custid, studid+custid +name, studid+custid+marks, studid+custid+name+marks

Employee table- empid, passport number, adhar num, pan num, empid+name, empid+name+address, passportnum+adhar number+name,.....

5. Unique key ---- is a column in which we need that data should be unique, duplicates are not allowed but it allows to keep the null values in the table

6. Foreign key ---- a column which is dependent on a column of other table(primary key) or same table(primary key) for integrity or correctness of data

a/c no	Balance	Type of Account	BranchCode	Custid
1	100000	Saving	000101	1000
2	200000	current	000101	1000
3	250000	Demat	000101	1000
4	230000	Saving	000101	1001
5	450000	Saving	000101	1002
6	450000	Saving	000101	

customer table

custid	Name	Mobile	Address	Email Address
--------	------	--------	---------	---------------

1000	Kishori	2222	Aundh	kk@gmail.com
1001	Revati	333	Aundh	<a href="mailto:Rev@gmail.com">Rev@gmail.com</a>
1002	Rajan	4444	Baner	<a href="mailto:Raj@gmail.com">Raj@gmail.com</a>

product

Prod id	Name	Price	catid
1	Lays	50	1
2	Coca cola	40	2
3	Maggi	60	3
4	Nachos	120	2

category

cid	Cname	Description
1	Chips	Crispy
2	Cold drink	Thanda Thanda cool coll
3	Snacks	Yummy

MySQL

SQL ----- Structured Query Language

PLSQL ----- Procedural Language SQL----→ If,loops, cursors

Types of statements

Types of statement		
DQL	Data Query Language--- it is used to retrieve data from table	select
DDL	Data definition language—are used to create table, modify table structure or delete table structure	Create,drop,truncate,alter
DML	Data manipulation Language-It allows to change the data	Insert, delete, update
DCL	Data control language-It allows to assign or remove permissions to users	Grant ,revoke
TCL	Transaction control statement	Commit, rollback , savepoint

step1:

Install mysql on your PC by using step by step procedure which is shared

Step2:



on windows start button search for mysql commandline client --→select it

it will open commandline window and ask for password

enter the password

will show

mysql>

Roomid	Rname	Room location	Facilities
123	Lotus	1 st floor	Ac/ tv/ sea facing
		1 st floor	Ac/ tv/ sea facing
			Ac/ tv/ sea facing

primary key ---→roomid

candidate keys ---→ Roomid, Rname

Alternate key ---→Rname

order table ----- amazon.com

Orderid	Itemid	qty	Date of purchase	custid	Adhar card	price
1	12	2	19-03-2021	100	12345	
1	13		19-03-2021	100		
1	20	2	19-03-2021	100		
	12	2	22-03-2021	100		

primary key ----- Orderid+itemid

candiate key ----→ Orderid+itemid

super key ---→ ordered+itemid+custid

flight ticket reservation system

Flight no	Seatno	Date of journey	Price	Passenger name	passengerid	
111111	1	12-03-2021		1		
111111		12-03-2021				

111111	1	13-03-2021		1		

flightno+seatno+date of journey

Flight no	Seatno	Date of journey	Price	departure	Passenger name	passengerid	
111111	1	12-03-2021		9:00 am	1		
111111		12-03-2021					
111111	1	13-03-2021		4:00pm	1		

Employee

Empid	Ename	deptno	Manager no	sal	Passport no
1	Rajesh	12	3	5678	345678
2	Kamal	null			null
3	Anil		10		
4					
10	Kishori				

dept

Deptno	Dname	location
12	HR	
13	Accounts	
14	admin	

To connect to mysql by different user

open cmd prompt

mysql -u user -p password

To create new database on your machine

mysql> create database iacsdmar22

mysql> create database if not exists iacsdmar22

To list all the databases

```
mysql>show databases;
```

To change to the database

```
mysql>use iacsdmar22;
```

To upload data from demobldmysql.sql. if file is loaded in the folder d:\data\ demobldmysql.sql

```
mysql>source d:\data\ demobldmysql.sql
```

To see the list of tables in the current database

```
mysql> show tables;
```

To delete the table

```
drop table emp;
```

```
drop table dept;
```

```
drop table salgrade;
```

To see the data in the table

```
mysql> select * from emp;
```

```
mysql>select * from dept;
```

```
mysql>select * from salgrade;
```

To see some selected fields

```
select empno,ename,sal
```

```
from emp;
```

To see the list of fields in the table

```
describe emp;
```

To see the table structure you can use describe

```
select <field list>|*
```

```
from <table name> |<list of table name>
```

```
where <condition>
```

Relation operators

<, >, >=, =, !=

Logical operators

and, or, not

To list all employees who works as clerk

```
select *
```

```
from emp
```

```
where job='clerk'
```

To find all employees with sal > 1500

```
select *
```

```
from emp
```

```
where sal>1500;
```

To find all employees who are working as clerk and sal >1000

```
select *
```

```
from emp
```

```
where job='CLERK' and sal > 1000;
```

to find all employees with sal >=1000 and sal <= 2000

```
select *
```

```
from emp
```

```
where sal>=1000 and sal<=2000;
```

When you check for range then use between ..... and operator

```
select *
```

```
from emp
```

```
where sal between 1000 and 2000;
```

to find all employees with sal not >=1000 and not <= 2000

```
select *
```

```
from emp
```

```
where sal not between 1000 and 2000;
```

select all employees who joined on '1981-02-20';

```
select *
```

```
from emp
```

```
where hiredate = '1981-02-20';
```

select all employees who joined between '1981-02-20' and '1981-11-17'

```
select *
```

```
from emp
```

```
where hiredate between '1981-02-20' and '1981-11-17';
```

select all employees who have not joined between '1981-02-20' and '1981-11-17'

```
select *
```

```
from emp
```

```
where hiredate between '1981-02-20' and '1981-11-17';
```

To find all employees with sal = either 1600 or 2850

```
select * from emp
```

```
-> where sal=1600 or sal=2850;
```

```
select *
```

```
from emp
```

```
where sal in (1600,2850,3000)
```

To find all employees who are working as clerk or manager

```
select *
```

```
from emp
```

where job in ('CLERK','MANAGER')

select \*

from emp

where name='james' and sal=2000;

and operator---→ second condition will be checked only if 1 st condition true

or operator---→ second condition will be checked only if 1 st condition false

not operator--→ it will negate the condition if the output of condition is true it will become false and if the output of condition is false it will become true

find all employee who do not earn any commission

select \*

from emp

where comm is null;

find all employee who earn some commission

select \*

from emp

where comm is not null and comm!=0;

to finds all employees with comm not 0 and not 300

select \*

from emp

where comm!=0 or comm!=300;

-----

select \*

from emp

where comm not in (0,300)

--	--	--

Between... and	To check the range of values, use between and operator The given values are inclusive	Select * from emp where sal between 1200 and 3000;
in	To check with multiple specific values with = in same column then use in operator	Select * from emp where sal in (1250,1600,3000)
Is null	To find all rows which has null values	Select * from emp where comm is null
Is not null	To find all rows which don't have null values	Select * from emp where comm is not null
Not between	To check the values out of range of values, use between and operator	Select * from emp where sal not between 1200 and 3000; Will display all row with sal <200 and sal >3000
Not In	To check with multiple specific values with not = in same column then use in operator	Select * from emp where sal not in (1250,1600,3000)
Like	To check pattern, use like operator	

all characters and dates has to be enclosed in single quote.

standard format for date ---- 'yyyy-mm-dd'

all relational operators, between... and ... , in operator can be used with date, numbers and characters

null is special value in database so always use is null or is not null operators with null values

but what if we don't know the exact value but we know the pattern

use like operator

to list all employees with name starts with J

select \*

from emp

where ename like 'J%'

