Types of database

1. Relational
    a. Structured database
    b. Data is stored in the form of tables
    c. Example – Oracle, MySQL, ProstgreSQL, SQL SERVER

2. NoSQL
    a. Unstructured
    b. Document , Key-value, Rowswise, columnwise format
    Examples: NoSQL,MongoDB,CouchbaseDB,Cassandra
3. GraphDB
    a. The data is stored as nodes and relations
    b. Can be displayed in the form of graphs
4. Memory DB
    a. The retrieval of data is very fast
    b. Stored in RAM, so temporary database
    c. Example: MemDB,VoltDB
5. Disk based database
    a. Access
    b. SQLite

MySQL

1. It is relational database
2. It stores data in structured manner
3. Sharing of data is possible
4. For security every user, username and password is given

- To create a database

Create Database test

- To change the database
  Use test

Banking table

acno	useid	name	email	mobile	balance	Adhar card	type	
100	1	Kishori	xxx@gmail.com	11111	3456	23456677	saving	
102	1	Kishori	xxx@gmail.com	11111	5555	23456677	current	
103	1	Kishori	asd@gmail.com	11111	3456	23456677	demat	
104	2	Rajan	raj@gmail.com	2222	345345	34234	saving	

useid	name	email	mobile	Adhar card	Pan card number
1000234	Kishori	asdf@gmail.co m	11111	23456677	111110
2000123	Rajan	rojrocks@gmail.com	2222	234534	2222
3034561	yash	yash@gmail.com	3333	44444	33333

acno	useid	balance	type	
100	1	3456	saving	
102	1	5555	current	
103	1	3456	demat	
104	2	345345	saving	

studid	courseid	cname	marks	
1	1	Java	99	
1	2	DBMS	100	
2	1	Java	97	
2	2	DBMS	99	

Room (Roomid,rname,location)

Primary key -→ Roomid

Superkey-→ roomid, romid+rname, roomid+location, roomid+rname+location

Customer(custid, cname,address,email)

Booking(roomid,custid,fromdate, todate, charges)

roomid	fromdate	todate	custid			
1	12-01-2008		1			
1	20-02-2009		1			

1. Primary key---- minimal number of the column or group of columns which identifies the row uniquely is called as primary key (unique + not null)

   For one table there can be only one primary key

   Primary key values should be unique and not null

   a. Simple primary key – It is formed by single column
   b. Composite primary key—It is formed by composite primary key

2. Super key------ any combination which identifies the row uniquely is called as super key
3. Foreign key-→ if one column references another column of same table or different table for correctness of data then it is called as foreign key
   Example

   In accounts table userid referenceces, user tables userid, then userid in user table has to be primary key.

   In one table there can be more than one foreign key.
   If a table has a composite primary key then a part of primary key can be a foreign key also.

   User table

useid	name	email	mobile	Adhar card	Pan card number
1000234	Kishori	asdf@gmail.co m	11111	23456677	111110
2000123	Rajan	rojrocks@gmail.com	11111	234534	2222
3034561	yash	yash@gmail.com	3333	44444	33333

Accounts table

acno	useid	balance	type	
100	1000255	3456	saving	
102	1000234	5555	current	
103	3034561	3456	demat	
104	2000123	345345	saving	

Employee

Empno	Name	Manager no	Job	sal
100	Kishori	102		
101	Rajan	102		
102	Anil	103		
103	Revati	101		

Stuid	Sname	mobile	adress
1	xxxx	34534	Ergdr
2	Yyyy	45645	dffg

Course

cid	cname	degree
12	JAVA	Java dude
13	Python	Python dude

Student-course

Primary key-→studid+courseid

Foreign key -→

Studid references Student(studentid)

Courseid references Course(cid)

studid	courseid	Date
1	12	
1	13	
2	12	

4. Candidate key--→ all possible combinations which are probable candidates to become primary key are called as candidate keys

useid	name	email	mobile	Adhar card	Pan card number
1000234	Kishori	asdf@gmail.co m	11111	23456677	111110
2000123	Rajan	rojrocks@gmail.com	11111	234534	2222
3034561	yash	yash@gmail.com	3333	44444	33333

   Example in above table
   Candidate keys ---→ userid, email, mobile, adhar card, pan card number

5. Unique key-----→ is any column in which the values should be unique then it is called as unique key
   In one table many unique keys can be there
   A unique key column can store many null values

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

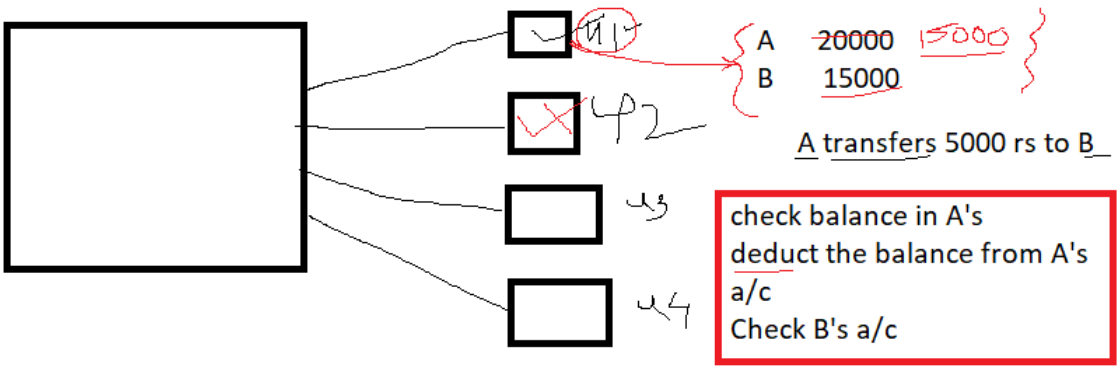
Kishori's balance ----- 150000

Rajan's balance -------350000

Durability

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

Isolation



To create database
Create database test;

Create database if not exists test;

To change database
Use test;

To display all table
Show tables

To see the column names and data type
desc emp;

To use database, we are using SQL and PLSQL

SQL -structured query language
PL-SQL ----- procedural language structured query language

categories		Statements
DQL	Data query language	select
DML	Data Manipulation Language	Insert, update, delete
TCL	Transaction control language	rollback, commit, savepoint
DCL	Data control language	grant , revoke
DDL	Data definition Language	Create, alter, truncate, drop

DQL --- select statement

1. Keywords are not case sensitive
2. Data is case sensitive
3. From statement is optional in mysql
   But compulsory in Oracle

Select [list of column name]
From [list of table names]
Where [condition]
Groupby list of columns
Having [aggregate column conditions]
Order by group of columns

user

useid	name	email	city	mobile	Adhar card	Pan card number
10234	Kishori	asdf@gmail.co m	pune	11111	23456677	111110
20123	Rajan	rojrocks@gmail.com	pune	11111	234534	
3034561	yash	yash@gmail.com	mumbai	3333	44444	

1. To display all the columns from user table

```
Select *
From user;
```

2. To display only userid, name, city

```
Select userid, name, city
From emp;
```

If you want to filter data based on condition, then use where clause

=	Equal to
!=	Not equal to
>, < , >=,<=	Relational operators
And, or, not	Logical opeartors

operators		
[not] in	When you want to check multiple values of one column with ,or condition then use in operator	Select * From user Where city in ('Pune','Mumbai');
[not] Between… and	When you want to check range of values then use between operator, the given values are inclusive	Select * From user Where userid between 10000 and 20000;
[not] like	To check the pattern, we can use like operator, and to design the pattern, use %- matches with 0 or more characters _ → matches with one character	
Is [not] null	To check value is null in any column then we use is null operator	

Note --- strings, date are enclosed in ' ' and number are written as is.

1. To display userid, name for all users who stays in pune city

```
Select userid,name
From user
Where city='pune';
```

2. To find all users with userid greater 15000

```
Select userid,name
From user
Where userid>15000;
```

3. To find all users who do not stay in pune

Select userid,name

From user

Where city!=’Pune’;

4. Find users who stays in either pune or Mumbai
   Select *
   From user
   Where city=’Pune’ or city=’Mumbai’

   Select *
   From user
   Where city in (‘Pune’,’Mumbai’);

5. Find users who do not stays in either pune or Mumbai

Select *

From user

Where city not in (‘Pune’,’Mumbai’);

6. To find all users whose Pancard number is not given
   Select *
   From user
   Where pancardnum is null;

7. To find all users whose Pancard number given
   Select *
   From user
   Where pancardnum is not null;

8. Find all users whose userid is > 10000 and <50000
   Select *
   From user
   Where userid >10000 and userid<50000

   Select *
   From user
   Where userid between 10001 and 49999

9. Find all users whose userid is <10000 or > 50000
   Select *
   From user
   Where userid not between 10001 and 49999

   To design patterns for like operator

Starts with s	Name like 'S%'	
Ends with e	Name like '%e'	
Somewhere i is there	Name like '%i%'	
i is at 2nd position and e is at the end	Name like '_i%e'	
E is at 3rd last position and starts with p	Name like 'p%e__'	

10. Find all user whose name starts with M
    Select *
    From user
    Where name like 'M%' or name like 'm%';

11. Find all user with name starts with N, ends with A and R at 3rd position
    Select *
    From user
    Name like 'N_R%A'

12. Find all user with name not starts with N
    Select *
    From user
    Where name not like 'N%'

13. Find all users with name starts with A and ends with e or p
    Select *
    From user
    Where name like 'A%e' or name like 'A%p'

14. Display all products with price greater than 40
    Select *
    from product
    where price > 40;
15. Display all products with qty > 10 and < 30

Select *

From product

Where qty between 11 and 29;

16. Display all products with catid is either 1 or 2
    Select *
    From product
    Where catid in(1,2)

17. Display all products with name starts with N
    Select *
    From product
    Where name like ‘N%’;
18. Display all products with price >= 40 and < =100
    Select *
    From product
    Where price between 40 and 100;

19. Display all products with name starts with N and ends with 4 or starts with M and ends with e

    Select *
    From product
    Where pname like ‘N%4’ or pname like ‘M%e’;

20. Display all products with price > 40 and < 100 and catid is either 1 or 2
    Select *
    from product
    where catid in (1,2) and price between 41 and 99;

vehicle : vid, name, make, model, chassie num
custid : custid, name, address, mobile, passportnum
veh-cust : custid, vid, date , price

Table name	primary	candidate	superkey	Foreign key	Unique key
Veh-cust	Custid+vid	Custid +vid, vid+date		Custid references customer(custid), vid references vehicle(vid)	
customer	custid	Custid, passportnum, mobile	All		Passport num, mobilenum
vehicle	vid	Vid, chassie num	Vid, chassie num, vid+name, Every possible combination of other columns with vid, Every possible		chassie num

			combination of other columns with chassie num		

1234 10

1234 12