Data base technologies

to store data in persistent storage device, then there are 2 styles

1. file----the data is store sequentially, and accessed also sequentially.
2. database—add, delete, modify and read data, becomes very easy as compared to file handling. hence most of places people prefer using databases, instead of files.

Types of databases

1.SQL databases

- data is stored as structured data.
- it is stored in table format
- we use SQL(structured query language) to manage data
- example—Oracle, mysql, PostgreSQL, SQL server
- these databases stores data table format, hence these are called as RDBMS

2. No SQL
    - data is stored in unstructured manner,
    - It is stored in json or document format
    - usually used in media application
    - examples : MongoDB,Cassandra,coutchbaseDB
3. GarphDB
    If you need to store and show the data in graph format
    examples: neo4j
4. Memory Database
    - In certain project data is very small, we need faster access to data, in such cases we use memory databases.
    - These databases are usually used in Research projects.
    - in these cases we need back up server also
    - examples : MemDB,VoltDB etc
5. disk based data base.
    - the data will be stored in the disk where the database is installed
    - examples: sqlite3, Access.

In SQL database to manage data the query language is used, Structured Query Language

Types of statements

DQL	data query language	select
DML	Data manipulation Language	insert, delete, update
DDL	Data definition language	create, alter, drop, truncate
DCL	Data control language	grant , revoke
TCL	transaction control language	commit, rollback, savepoint

Uses of databases

1. reduces redundancy of data
2. sharing of data is possible.
3. availability of data is increased.
4. accessibility of the data is increased.

student

sid	sname	address	mobile
1	Rajat	Kothrud	44444
2	Rakesh	Akurdi	5545454
3	Rekha	Baner	55555
4	Sharmila	Kothrud	55556666

To retrieve all rows

select * from student;

to find all students who stays in Kothrud

select * from student where address='Kothrud'

Keys in databases

1. Primary key: minimal set of columns which identifies the row uniquely , are called as primary key.
   student
   corse
   student-marks

sid	cname	marks1	exam date	marks2
1	java	99		97
1	c++	88		97
2	java	55		78
2	c++	99		34

Room

roomid	customerid		booking date	rate of booking	mobile number	saleman no
100	1		2 jan 24	5000	44444	10
100	1		4 jan 24	5000	44444	10
101	1		2 jan 24			

movie management

movie num	screen num	date of booking	seat no	show time	price
1	1	1 jan 24	1		

1	1	2 nd jan	1		

empno	ename	email	mobile	adharnum	passport num	pannumer	salary	desg

2. Candidate key:

All possible minimal combinations, that may become primary key, are called as candidate key,

Out of these one chosen will become primary key

and all remaining will become alternate key.

3. unique key--- any single column, which has unique values, is called as unique key

it can contain more than one null values, but not null values will be unique.

4. foreign key--- if we want to enter valid data in a column, and for that we may need to refer primary key of other table, or primary key of same table, then it is called as foreign key

5. Alternate key

all candidate keys, which are not selected as primary key are called as alternate key

6. super key

any combination of the columns which identifies the uniquely is called as super key

to install mysql
https://dev.mysql.com/downloads/installer/

using emp , dept

1. to list all the employees with job clerk
   select * from emp where job='CLERK'
2. to list all employees with job clerk or sal >2000
   select * from emp where job='CLERK' or sal >2000;
3. to list all employees with job manager and sal >2000
   select * from emp where job='CLERK' and sal >2000;
4. To list all employees with hiredate '1982-12-09'
   select * from emp

   where hiredate='1982-12-09';
5. To list all employees with sal < 3000 and name =smith
   select * from emp
   where sal < 3000 and name='smith';
6. To list all employees with sal >= 1300 and <=3000

select * from emp where sal between 1300 and 3000

operators we can use in database.

Arithmetic operators

+, -, *, / , %

Logical operators

and, or, not

Relational operators

<, >, <=, >=, !=, =

[not] between ...and	whenever you want to check range of values, the use between...and operator, the values we use are inclusive	select * from emp where sal between 1300 and 3000
[not] in	when you want to check multiple values in single column, then use in	select * from emp where sal in(1300,2000,3000)
is [not] null	this operator will check for the null values	select * from emp where comm is null;
[not] like	like operators will allow you to design patterns to design patterns we use % and _ % matches with 0 or more characters _ matches with 1 character	select * from emp where ename like ‘A%’ select * from emp where ename like ‘_A%’

7. To list all employees with sal not > 1300 and <3000
   select * from emp where sal not between 1300 and 3000

8. to list all employees joined in jan-81
   select * from emp where hiredate between ‘1981-01-01’ and ‘1981-01-31’

9. to list all employees joined in year 1981
   select * from emp where hiredate between ‘1981-01-01’ and ‘1981-12-31’

10. to list all employees with sal is either 1300, 2500 or 3000
    select * from emp where sal in (1300,2500,300)
11. to list all employees working as either clerk or manager or analyst
    select * from emp where job in (‘CLERK’,’MANAGER’,’ANALYST’)
12. to list all employees not working as either clerk or manager or analyst

select * from emp where job not in ('CLERK','MANAGER','ANALYST');

13. To list all employee with comm is null
    select * from emp
    where comm is null;
14. To list all employees with name starts with either m or starts with J
    select * from emp where ename like ‘M%’ or ename like ‘J%’
15. list all employees with name starts with A and E at second last position.
    select * from emp where ename like ‘A%E_’
16. list all employees with name starts with either A or M and ends with either N or R
    select * from emp where ename like ‘A%N’ or ename like ‘M%R’ or ename like 'A%R’ or ename like 'M%N
17. list all employees with either N at second position or N at third position.
    select * from emp where ename like ‘_N%’ or ename like ‘__N%’

REGEXP

.	to match any one character
[a-zA-Z]	it matches with alphabets
[0-9]	it matches with any one digit
*	matches the preceding pattern for 0 or more times
+	matches the preceding pattern for 1 or more times
?	matches the preceding pattern for 0 or one times
{m}	exactly m occurrences
{m,n}	it matches with minimum m and maximum n occurrences
{m,}	it matches with minimum m and maximum any number of occurrences
^	it matches the pattern at the beginning of the string
$	it matches the pattern at the end of the string
[^a-z]	it matches with any character except a-z
(abc|pqr|mnx)	it matches with any one pattern abc or pqr or mnx

1. find all enames that contains A somewhere in the name
   select * from emp where ename like ‘%A%’
   select * from emp where ename REGEXP ‘A'

2. find all enames that contains A at the beginning in the name
   select * from emp where ename like ‘A%’
   select * from emp where ename REGEXP ‘^A'
3. find all enames that contains A at the end in the name
   select * from emp where ename like ‘%A’
   select * from emp where ename REGEXP ‘A$'

4. To list all employees with name starts with either m or starts with J
   select * from emp where ename like ‘M%’ or ename like ‘J%’
   select * from emp where ename REGEXP ‘^[MJ]’

5. list all employees with name starts with A and E at second last position.
   select * from emp where ename like ‘A%E_’
   select * from emp where ename regexp '^A.*E.$’

6. list all employees with name starts with either A or M and ends with either N or R
   select * from emp where ename like ‘A%N’ or ename like ‘M%R’ or ename like 'A%R’ or ename like 'M%N
   select * from emp where ename regexp ‘^[AM].*[NR]$’;

7. list all employees with either N at second position or N at third position.
   select * from emp where ename like ‘_N%’ or ename like ‘__N%’
   select * from emp where ename regexp ‘^.N|^..N’
   select * from emp where ename regexp ‘^..?N’
8. List all employees with name starts with A and ends with N or Starts with M and ends with R
   select * from emp where ename regexp ‘^A.*N$|^M.*R$’
9. List all employees with name not starting with A and ends with N or not Starts with M and ends with R
   select * from emp where ename not like ‘A%N’ or ename not like ‘M%R’
   select * from emp where ename regexp ‘^[^A].*[^N]$|^[^M].*[^R]$’

10.