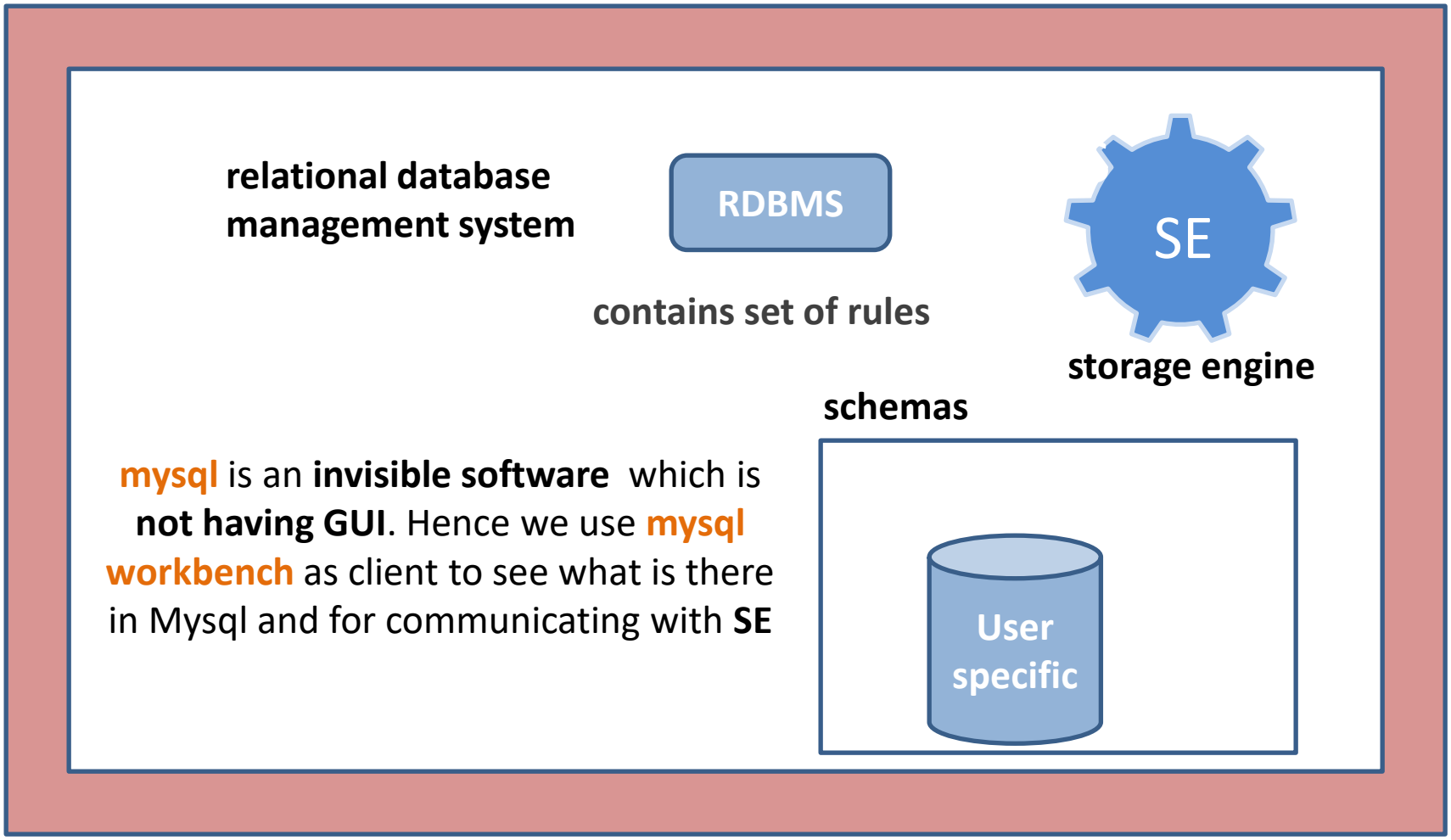
MYSQL

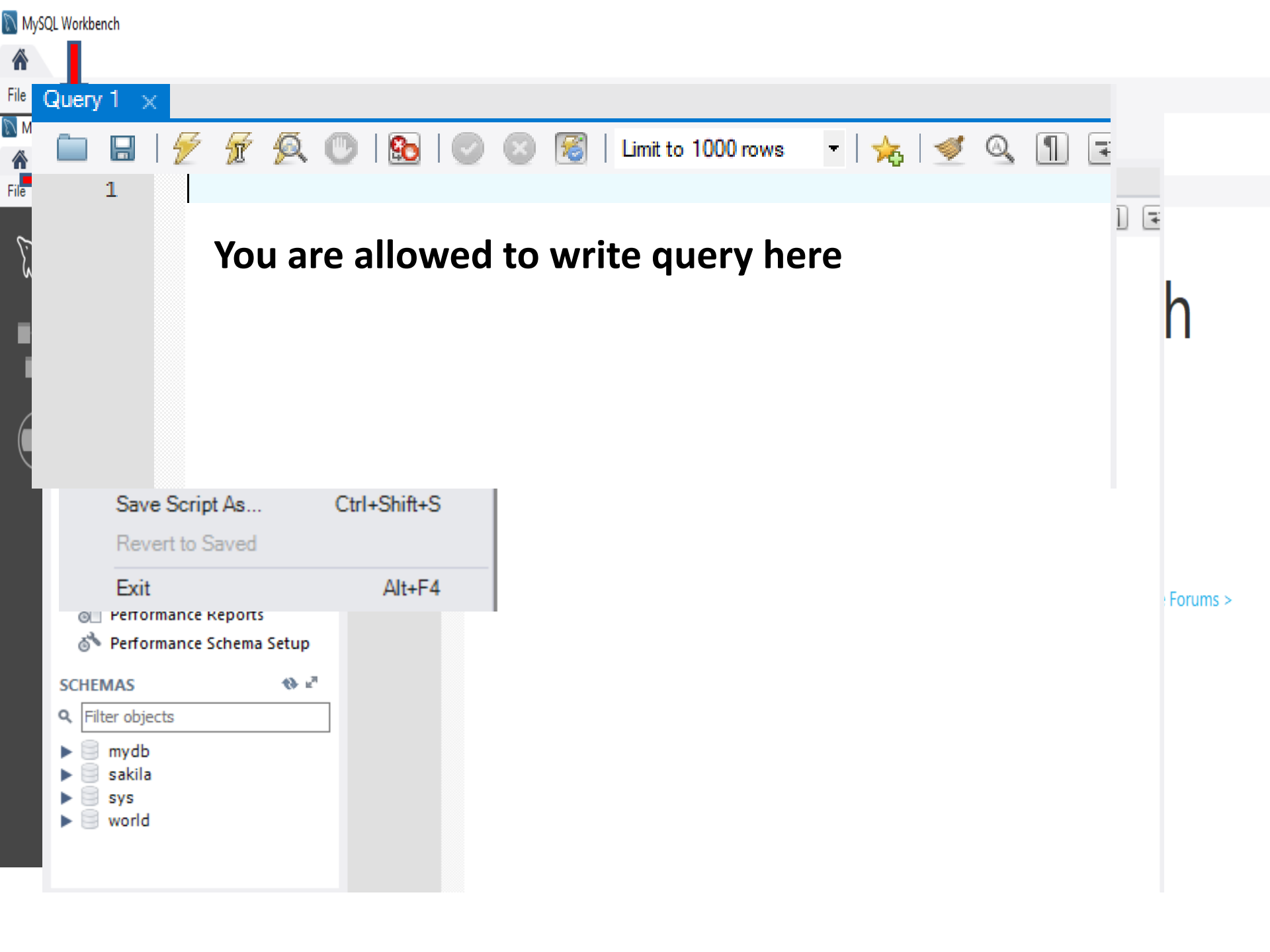
This Course will give In-depth Information related to MY-SQL language

Author: Lokanatha Palle

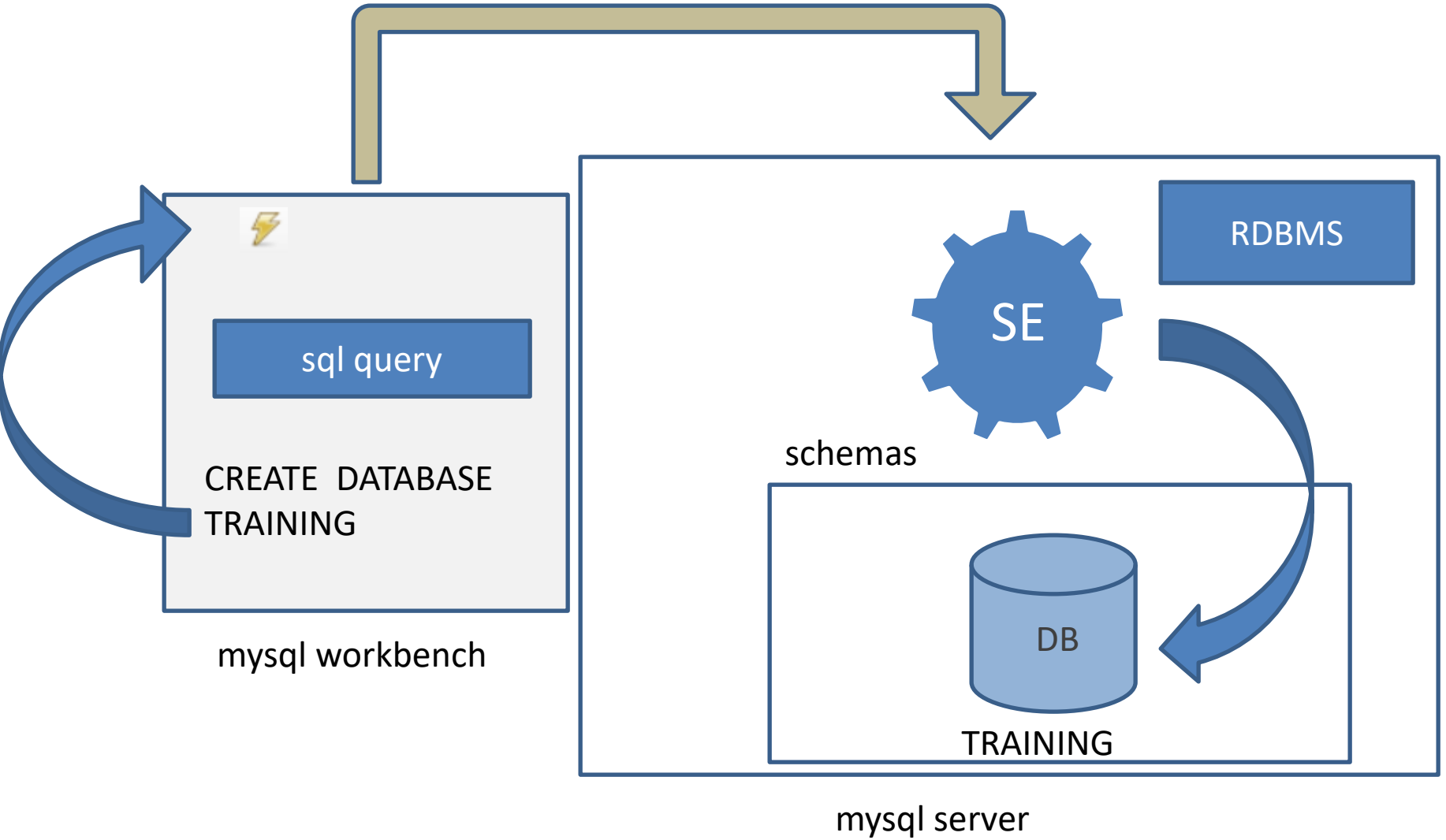
Date: 12-Dec-2017

mysql overview




MySQL Workbench
File
Query 1
Limit to 1000 rows
1
You are allowed to write query here
Save Script As... Ctrl+Shift+S
Revert to Saved
Exit Alt+F4
Performance Reports
Performance Schema Setup
SCHEMAS
Filter objects
mydb
sakila
sys
world
Forums >

mysqlworkbench and mysql communication



data types in sql

- int
- smallint
- bigint
- tinyint
- mediumint
- decimal(p,d)
- float
- double

- char
- varchar
- binary
- varbinary
- datetime
- date
- time
- timestamp
- year

- bit
- real
- numeric(p,d)
- xml
- text
- tinytext
- mediumtext
- longtext
- blob

note : decimal and numeric can have upto p digits with d decimals

sizes of data types 1

Datatypes	Size in bytes
• tinyint • smallint • mediumint • int • bigint • decimal(p,d)	1 byte 2 bytes 3 bytes 4 bytes 8 bytes can store upto 65 digits 30 can be after decimal point
• bit	1 bit (can store either 0 or1) note: usually used to store true or false

sizes of data types 2

Data types	Size in bytes
• char(x)	can store max 255 characters
• varchar(x)	can store upto 16383 characters
• tinytext	can store upto 256 bytes
• text	can store upto 65535 bytes (~64kb)
• mediumtext	can store upto 16777215 bytes (~16mb)
• longtext	can sore upto 4294967295 bytes (~ 4gb)

char vs varchar

char

- can store upto 255 characters
- fixed size datatype

varchar

can store upto 16383 characters

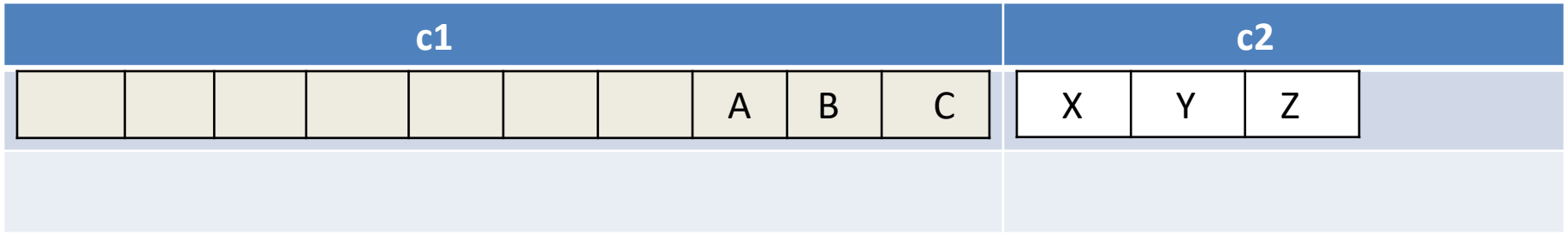
varying size datatype

example

```
create table sample
(
    c1 char(10),
    c2 varchar(10)
)
```

```
insert into sample values('ABC','XYZ')
```

sample



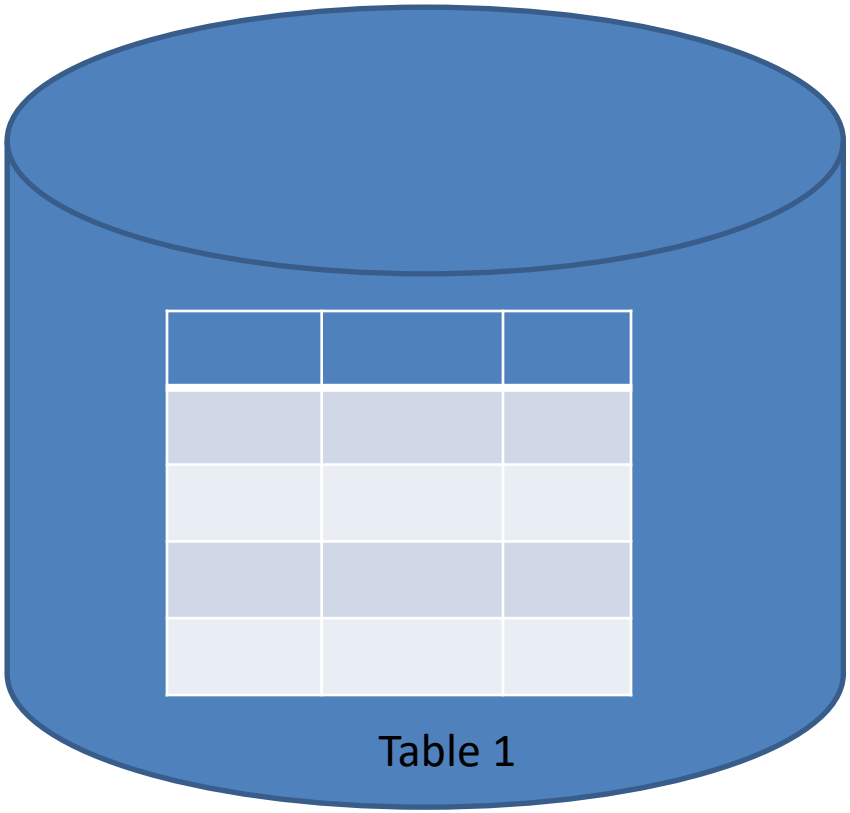
table

- tables are the combination of rows and columns

  table = row+column

- rows = tuples
- columns = attributes

  or properties



creating table

syntax

```
create     table     table_name
(
   column1      datatype1[(size)]  [constraint],
   column2      datatype2,
   column3      datatype3,
    ……..
)
```

note: as per industry standards table names and column names must not be plurals.

table sample

req:- create a student table to store students details sid,name,class,dob

Int	vc(40)	vc(40)	date
sid	**name**	**class**	**dob**
1	Rajeev	9th class	1999-10-23
2	Veena	10th class	1998-05-30

student

```
create table student(
sid int,
name varchar(40),
class varchar(40),
dob date
)
```

```
insert into student values (1,'Rajeev', '9th class','1999-10-23')

insert into student values (2, 'Veena', '10th class', '1998-05-30')
```

note : date should be in **'yyyy-mm-dd'** format

table creation assignment

- create a table with the name product and with column names (**pid**→int, **pname**→varchar(40),**cost**→int, **manufacturer_name**→varchar(40), **manufactured_date**→date)
- the table must be created in PalleTraining DB (assuming that the DB is already available in your sql server).
- insert the following data into product table.

  1, lux, 34, HUL, dec-12-2017

  2, locks, 1200, Godrej, Jan-11-2018
- write query for displaying data present in product table

constraints

- using constraints we can limit the data which is **coming** into table columns.
- sql supports following constraints

default	not null	check
primary key	unique	foreign key

default constraint

default constraint is **used** for inserting default value into column when user not supplying value

```
create table student
(
   sid int,
   name varchar(40),
   city varchar(40) default 'bangalore'
)
```

Default
banglore

sid	name	city
101	Raj	chennai
103	Veena	bangalore

```
insert into student values(101,'Raj','chennai')

insert into student(sid,name) values(103,'Veena')
```

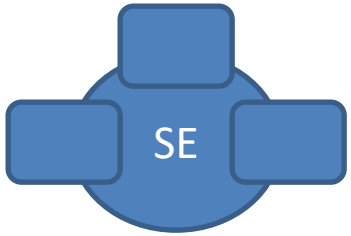
notnull constraint

when you set the not null constraint to a specific column , that column will **not** allow null value

```
create table employee
(
  Eid int  not null,
  Name varchar(40),
  Email varchar(40)
)
```

NOT NULL

Eid	Name	Email
1	ravi	r@gmail.com
2	NULL	NULL




```
insert into employee values (1,'ravi','r@gmail.com')
insert into employee values (2,null,null)
insert into employee values (null,'suresh','s@gmail.com')
```

Not null constraint will not allow null values.

check constraint

using check constraints we can limit the **range of** permissible values into a specific column

req: create a table with employee details eid,name,age_in_years (age must be between 18 to 60)

```
create table employee(
eid int,
name varchar(40),
Age_in_yrs int check(Age_in_yrs between 18 and 60)
)
insert into employee values(103,'Madhav',17);
```

Do You think that SE will accept this command?

18 - 60

ne | Age_in_yrs

SE will not Accept this Command

eid	name	Age_in_yrs
101	Raj	25

Error

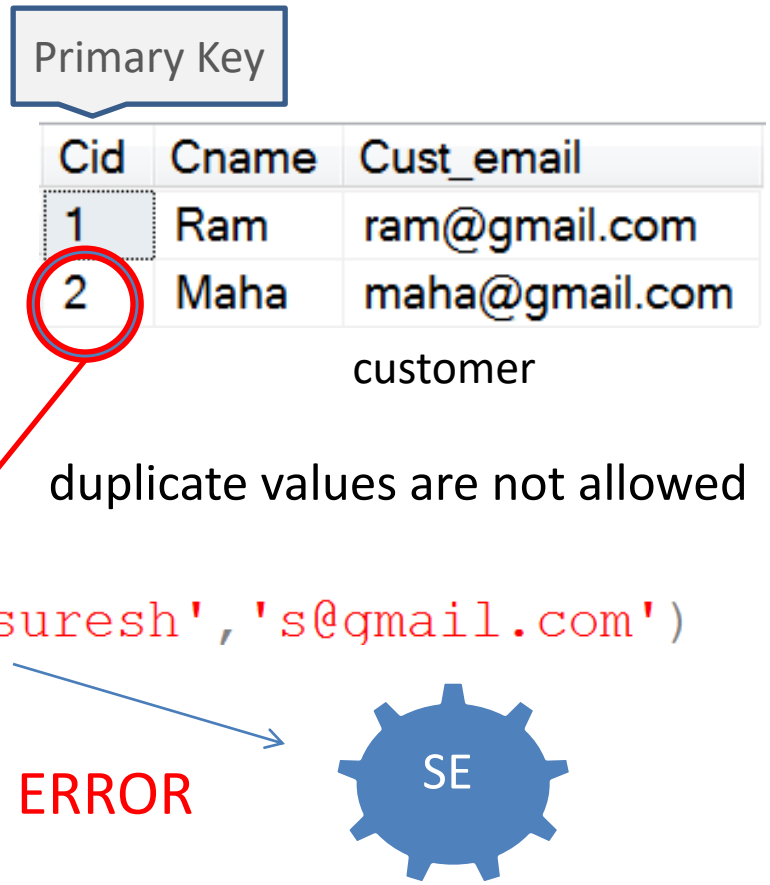
whether SE will accept this values?

primary key

- primary key **gives** uniqueness to the tables **rows**
- **only** one primary key is allowed **per table**
- primary key will **not** allow **null** values and duplicate values

sample

```
create table customer
(
  Cid int primary key,
  Cname varchar(40),
  Cust_email varchar(50)
)

insert into customer values (2,'suresh','s@gmail.com')
```



unique

- unique constraint **gives uniqueness** to the tables rows
- **multiple** unique constraints are **allowed** per table
- unique contraint will **not allow duplicate values** and **allows null values**

```
create table customer
(
    c_id int unique,
    name varchar(40),
    cell_no bigint unique,
    product varchar(40)
)
```

Unique

int varchar(40) bigint varchar(40)

c_id	name	cell_no	product
101	Ram	9943300821	toothbrush
102	Mahesh	NULL	soap
103	Ram	NULL	lux

customer

```
insert into customer values (101,'Mahi',9865226020,'shampoo')

insert into customer values (103,'Ram',null,'lux')
```

ERROR

composite primary key

when we apply primary key constraint on **more than one column**, then it is called composite primary key.

```
create table customer
(
    name varchar(40),
    cell_no varchar(20),
    product varchar(40),
    constraint cpk
    primary key (name,cell_no)
)
```

primary key

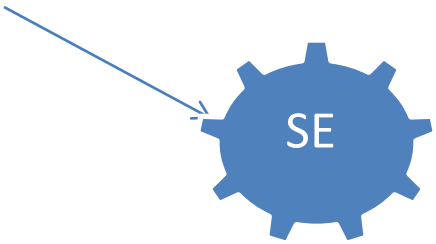
varchar(40) varchar(20) varchar(40)

name	cell_no	product
Ravi	+91-9809590910	toothbrush
mahesh	+91-8820170898	lux soap
Ravi	+91-9810820188	lion dates

```
insert into customer values('Ravi','+91-9809590910','lux')

insert into customer values('Ravi','+91-9810820188','lion dates')
```

how many PK's are created in the table

Only 1 PK for 2 columns.



ERROR

primary key vs unique

primary key

- only one primary key is allowed per table
- will not allow NULL values
- it will internally create clustered index.(we will understand later)

unique

- any number of unique constraints are allowed/table
- will allow NULL values
- unique will create non clustered index.

constraints assignment

int	varchar(40)	int	varchar(10)	int	varchar(40)
eid	name	salary	bg	age	email
4	ravi	36000	O+ve	48	ravi@gmail.com
6	suresh	38000	O+ve	56	null

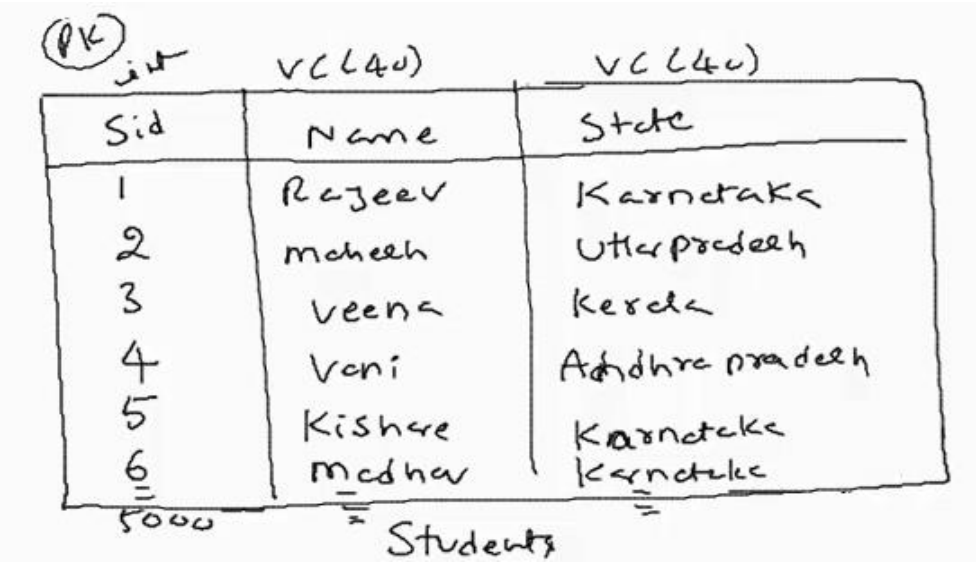
req:

1. eid column must not allow any duplicate or null values
2. name column must not allow any null values
3. bg should have default value as o+ve
4. age should have range from 18-60 years
5. email column should allow null values and no duplicate values

normalization

- using normalization we can **reduce** the data duplication or data redundancy.
- usually normalization process involves splitting a single table into multiple tables.
- It is **recommended** to create a new table for **storing** predictable repeating data .
- Ex: blood group names, state names / province names in a country…….
- normalization is used for avoiding the insert/update/delete anomaly or inconsistency

Consider this **student** table for understanding normalization



- Assume this table consist of 5000 records
- Observe the table and tell me , is there any **data duplication** (repetition of same data)

The **state column** data are duplicated, and tell me is state column data are **predictable**?

Definitely the state column data are predictable data, because we have only 29 states in our country. All the 5000 students must belong to any 1 of 29 states.

Memory required for storing 1 state name is,

Then for storing all the state names how much memory is required ??

Very huge memory is required

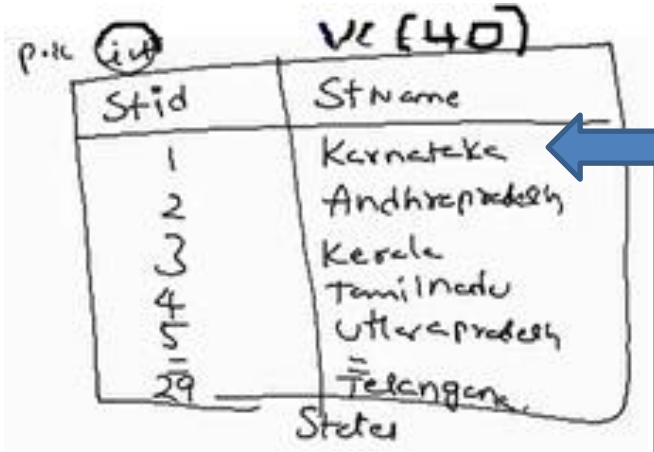
2000 → Karnataka → 9 Char's → 2 byte x 9 = 18 byte

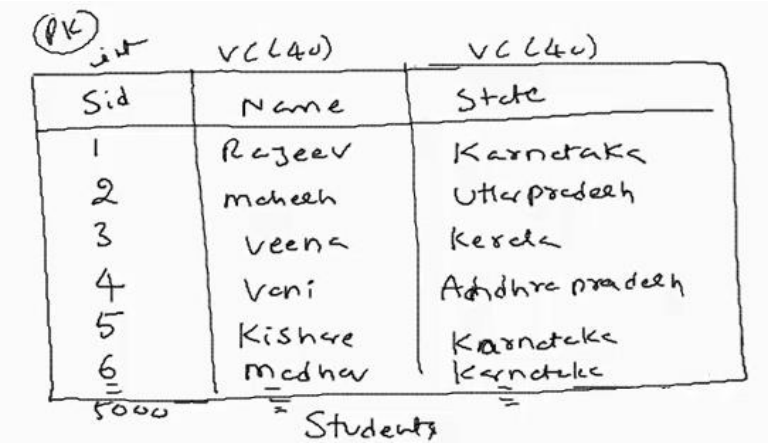
2000 x 18 ⇒ 36000 byte

How can we avoid this duplication,

By splitting the single table into two tables

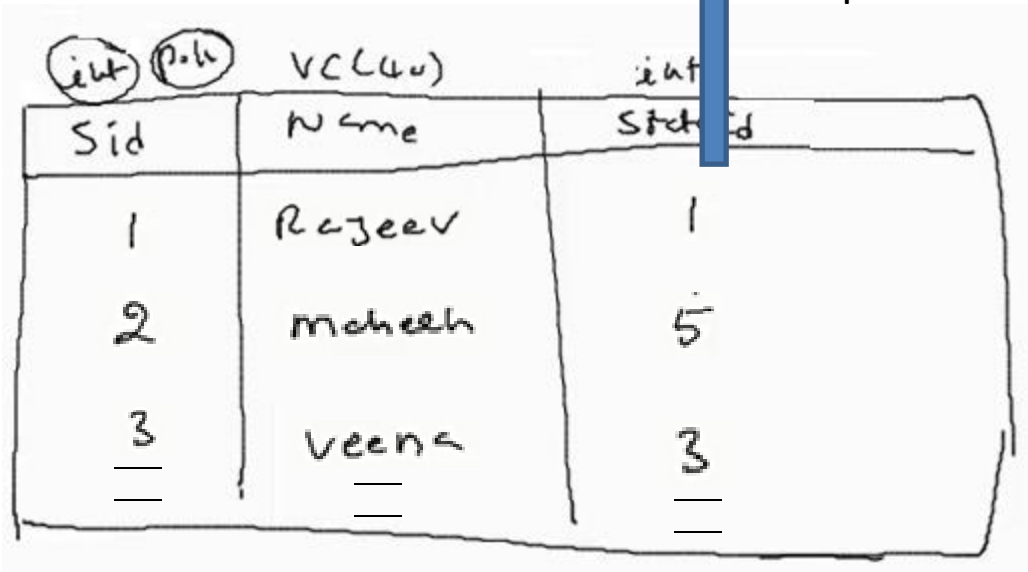
Now i have created 1 separate table for storing all the state names, and I have given a unique id For each state





UnNormalized table

Now I will create a student table and in place of state column , I will give the state id.

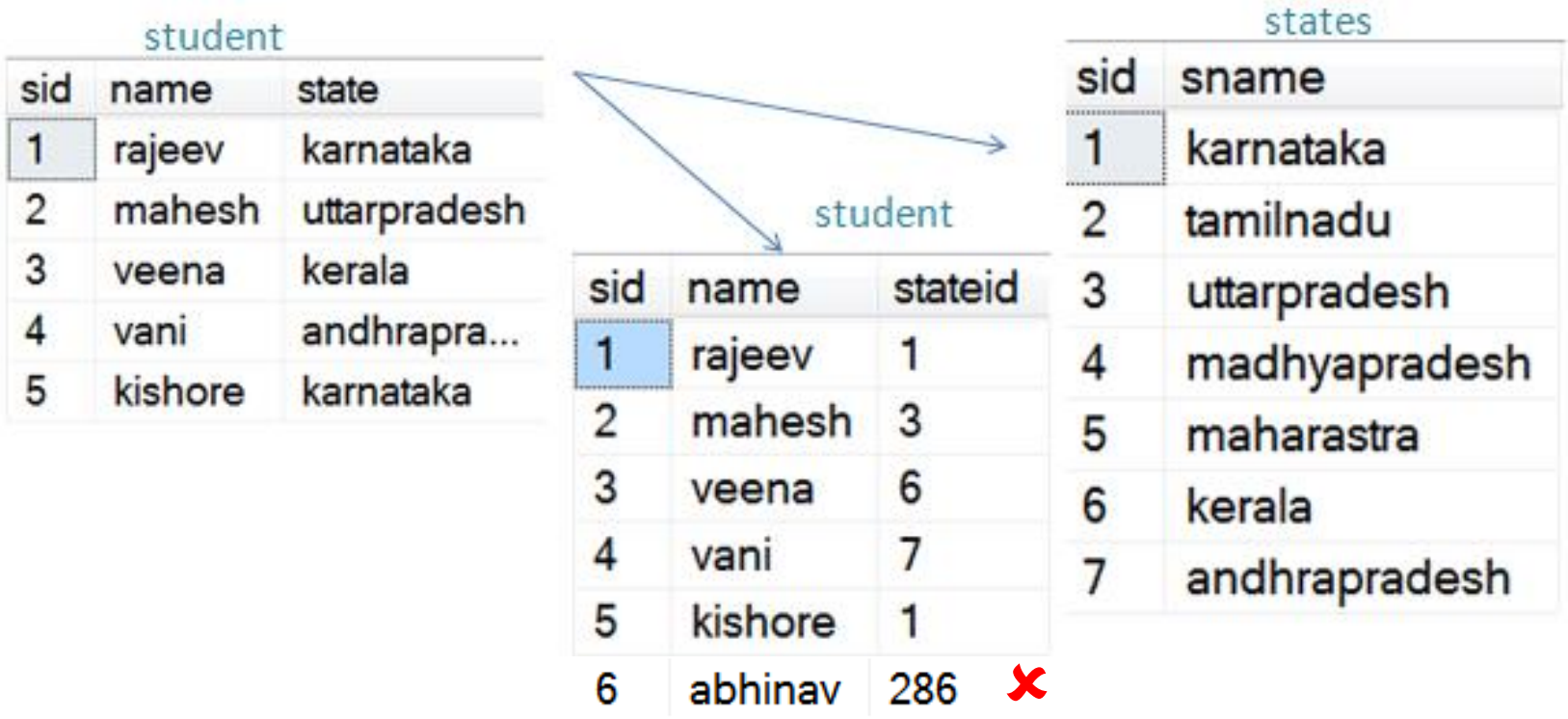


We have reduced the data duplication by using the **normalization** technique

foreign key constraint

- using **foreign key** constraint we link / **relate** 2 or more **tables**.
- using **foreign key** we can achieve referential **integrity**

  When we are normalizing the tables there are chances for getting insert and update anomaly



student

sid	name	state
1	rajeev	karnataka
2	mahesh	uttarpradesh
3	veena	kerala
4	vani	andhrapra...
5	kishore	karnataka

student

sid	name	stateid
1	rajeev	1
2	mahesh	3
3	veena	6
4	vani	7
5	kishore	1
6	abhinav	286 ✘

states

sid	sname
1	karnataka
2	tamilnadu
3	uttarpradesh
4	madhyapradesh
5	maharastra
6	kerala
7	andhrapradesh

```
insert into student values(6,'abhinav',286)
```

To eliminate this problem, we will be using Foreign key constraint

- foreign key will allow null values
- multiple foreign key constraints can be created per table
- any foreign key column mapped to other column from any table must be defined with either primary key or unique constraint

syntax

```
create  table  table_name
(
   column1  datatype ,
   column2  datatype,
   foreign key (column name)
   references  table_name (column name)
)
```

Example

```
create table state
(
    sid int primary key,
    statename varchar(40)
)
create table student
(
    sid int,
    name varchar(40),
    stateid int,
    foreign key (stateid) references state(sid)
)
```

Primary key

state

sid	statename
1	karnataka
2	tamilnadu
3	uttarpradesh
4	madhyapradesh
5	maharastra
6	kerala
7	andhrapradesh

How will you link the columns

sid	name	stateid
1	rajeev	1
2	mahesh	3
3	veena	6
4	vani	7
5	kishore	1

student

ERROR

```
insert into student values(6,'abhinav',286)
```

normalization lab1

- normalize the employee table use foreign key

eid	ename	age	bg	state
1	madhesh	25	o+ve	karnataka
2	suresh	24	o-ve	andhrapradesh
3	veena	30	o-ve	karnataka
4	kiran	35	b+ve	karnataka
5	ravi	25	b+ve	kerala

lab 1 solution

Blood_group

primary key

id	bg
1	o+ve
2	o-ve
3	b+ve
4	b-ve

state

primary key

state_id	state_name
1	karnataka
2	tamilnadu
3	california
4	andhrapradesh
5	kerala

Foreignkey

Foreignkey

employee

eid	ename	age	bg	state
1	madhesh	25	o+ve	karnataka
2	suresh	24	o-ve	andhrapradesh
3	veena	30	o-ve	karnataka
4	kiran	35	b+ve	karnataka
5	ravi	25	b+ve	kerala

Un-normalized employee table

New_employee

eid	ename	age	bg	state
1	madhesh	25	1	1
2	suresh	24	2	4
3	veena	30	2	1
4	kiran	35	3	1
5	ravi	25	3	5

Normalized employee table

normalization lab 2

- normalize the student table and use foreign key

id	name	city	state	country
1	kumar	banglore	karnataka	india
2	mahesh	mysore	karnataka	india
3	vishnu	banglore	karnataka	india
4	paul	San Franscisco	California	us
5	basker	chennai	tamilnadu	india

wrong solution

Why?

cid	c_name
1	banglore
2	chennai
3	mysore
4	San Franscisco
5	hyderabad

state_id	state_name
1	karnataka
2	tamilnadu
3	california
4	andhrapradesh
5	kerala

country_id	country_name
1	india
2	united states

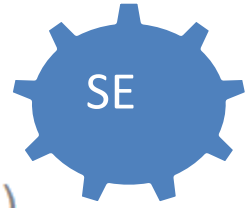
F.K

F.K

F.K

id	name	city_id	state_id	country_id
1	kumar	1	1	1
2	mahesh	3	2	1
3	vishnu	1	1	1
4	paul	4	3	2
5	baskar	2	4	1
6	Suresh	1	2	2

Is it a valid data?



Assume we are trying to insert

```
insert into student values(6,'Suresh',1,2,2)
```

Normalized student table

solution

Un-normalized student table

student

id	name	city_id
1	kumar	1
2	mahesh	3
3	vishnu	1
4	paul	4
5	baskar	2

student

id	name	city	state	country
1	kumar	banglore	karnataka	india
2	mahesh	mysore	karnataka	india
3	vishnu	banglore	karnataka	india
4	paul	San Franscisco	California	us
5	basker	chennai	tamilnadu	india

F.K

F.K

F.K

cid	c_name	state_id
1	banglore	1
2	chennai	2
3	mysore	1
4	hyderabad	4

city

sid	statename	country_id
1	karnataka	1
2	tamilnadu	1
3	andhrapadesh	1
4	california	2
5	kerala	1

state

country_id	country_name
1	india
2	united states

country

types of sql statements

- sql statements **categorized** into
- dml statements(insert, update, delete)
- ddl statements (create, alter, drop, truncate)
- dql statements (select)
- tcl statements (commit, rollback)
- dcl statements (grant, revoke used only by dba not by db programmers)

generic select statement

```
select select_list
 from table_list
[ where search_conditions ]
[ group by group_by_list ]
[ having search_conditions ]
[ order by order_list [ asc | desc ]  ]
```

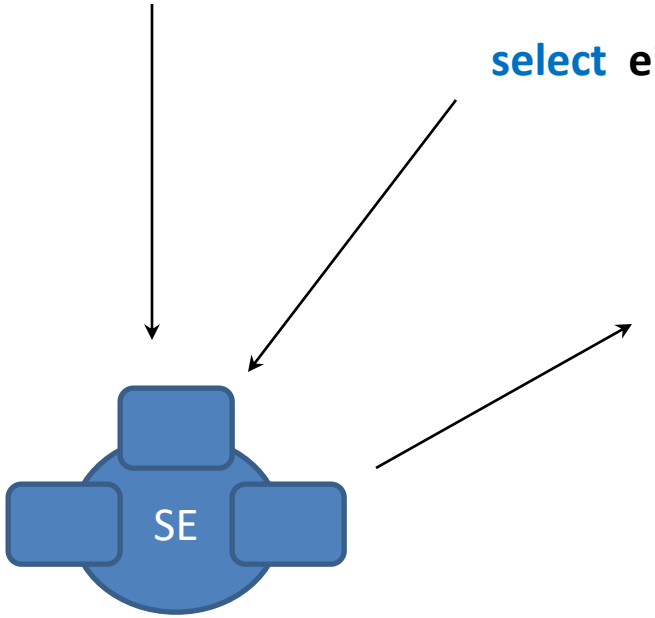
select sample 1

display all employee details.

employee

eid	fname	lname	age	salary	dept	doj
1	rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
3	kishore	kumar	27	36000	android	2011-10-16
4	abimanyu	biswal	22	NULL	android	2010-02-20
5	soni	kumar	24	21800	.net	2009-06-21
6	anu	_singh	22	12000	db	2010-10-23
7	_dinesh	moh%anty	23	15000	.net	2009-08-26
8	nishala	_kumari	22	18000	db	2008-07-19
1	rajeev	sukla	23	12000	.net	2011-10-23

select * from employee



select eid,fname,lname,age,salary,dept,doj from employee

eid	fname	lname	age	salary	dept	doj
1	rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
3	kishore	kumar	27	36000	android	2011-10-16
4	abimanyu	biswal	22	NULL	android	2010-02-20
5	soni	kumar	24	21800	.net	2009-06-21
6	anu	_singh	22	12000	db	2010-10-23
7	_dinesh	moh%anty	23	15000	.net	2009-08-26
8	nishala	_kumari	22	18000	db	2008-07-19
1	rajeev	sukla	23	12000	.net	2011-10-23

students must copy the table data in their note.

Employee

eid	fname	lname	age	salary	dept	doj
1	rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
3	kishore	kumar	27	36000	android	2011-10-16
4	abimanyu	biswal	22	NULL	android	2010-02-20
5	soni	kumar	24	21800	.net	2009-06-21
6	anu	_singh	22	12000	db	2010-10-23
7	_dinesh	moh%anty	23	15000	.net	2009-08-26
8	nishala	_kumari	22	18000	db	2008-07-19
1	rajeev	sukla	23	12000	.net	2011-10-23

select sample-2

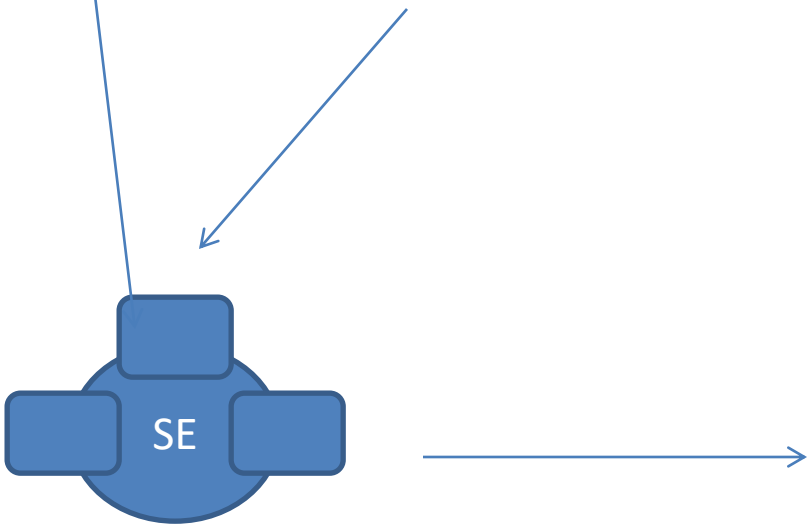
write a query for producing the following result set from employee table

ERS

Fullname	age
rajeevsukla	23
sowmyakumari	23
kishorekumar	27
abimanyubiswal	22

select concat(fname,lname),age from employee

select concat(fname,lname) as 'Fullname',age from employee



Fullname / concat(fname,lname)	age
rajeevsukla	23
sowmyakumari	23
kishorekumar	27
abimanyubiswal	22
———	———

student

all students are expected to write the patient table along with the data shown in the next screen since for all queries assignment we use same table

table : patient

pid→ int

fname→varchar(40)

lname→varchar(40)

age→int

bg→varchar(40)

- table must be created without any constraints

Note: create this table in the last pages of your **note book** as this table used for all assignments in in my-sql.

pid	fname	lname	age	bg
1	madhava	reddy	45	o+ve
2	abhinav	bandra	45	o-ve
4	hari	kiran	60	b-ve
3	madhava	kiran	52	o+ve
5	veena	kumari	42	NULL
6	k_iran	kumar	39	b-ve
2	abhinav	bandra	45	o-ve
7	mahes%h	nambootri	36	b+ve
8	rahul	kumar	46	b-ve
9	bharat	kumar	56	b-ve

select statement lab-1

- write a query for displaying all data present in patient table (using *)
- write a query for displaying all columns data present in patient table without using *
- write a query for displaying all patients fullnames, pid, age

select statement lab-2

- Write a query for displaying all patients full names along with the ages by incrementing all patients age by 2 years (Sample ERS)

fullname	age
madhavareddy	47
abhinavbandra	47
harikiran	62
_____	___
_____	___

select statement lab-3

- write a select statement for displaying the following result set?

patient details
madhava reddy's bg is o+ve and he/she is from India
abhinav bandra's bg is o-ve and he/she is from India
hari kiran's bg is b-ve and he/she is from India
madhava kiran's bg is o+ve and he/she is from India
NULL
k_iran kumar's bg is b-ve and he/she is from India
abhinav bandra's bg is o-ve and he/she is from India
mahes%h nambootri's bg is b+ve and he/she is fro...
rahul kumar's bg is b-ve and he/she is from India
bharat kumar's bg is b-ve and he/she is from India

order by clause

- used to **order data** present **in** a **table** based on **one or more columns**.
- use **ASC** keyword for ascending order and **DESC** for descending order.
- default order is ASC
- syntax:

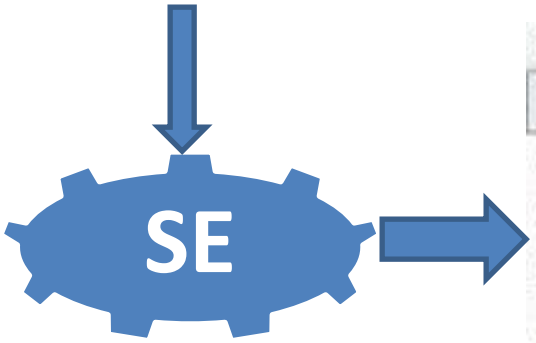
```
select select_list from table_ name
order by c1 asc/desc,c2  asc/desc,..
```

employees table

int	vc(40)	vc(40)	int	int	vc(40)	date
eid	fname	lname	age	salary	dept	doj
1	rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
3	kishore	kumar	27	36000	android	2011-10-16
4	abimanyu	biswal	22	NULL	android	2010-02-20
5	soni	kumar	24	21800	.net	2009-06-21
6	anu	_singh	22	12000	db	2010-10-23
7	_dinesh	moh%anty	23	15000	.net	2009-08-26
8	nishala	_kumari	22	18000	db	2008-07-19
1	rajeev	sukla	23	12000	.net	2011-10-23

display employees in ascending order of their fname.

select * from employees order by fname



final result set

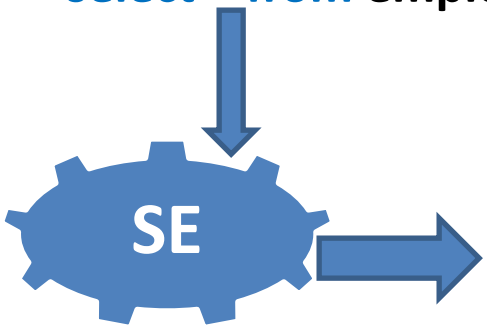
eid	fname	lname	age	salary	dept	doj
7	_dinesh	moh%anty	23	15000	.net	2009-08-26
4	abimanyu	biswal	22	NULL	android	2010-02-20
6	anu	_singh	22	12000	db	2010-10-23
3	kishore	kumar	27	36000	android	2011-10-16
8	nishala	_kumari	22	18000	db	2008-07-19
1	rajeev	sukla	23	12000	.net	2011-10-23
1	rajeev	sukla	23	12000	.net	2011-10-23
5	soni	kumar	24	21800	.net	2009-06-21
2	sowmya	kumari	23	19000	db	2010-11-13

employees table

int	varchar(40)	varchar(40)	int	int	varchar(40)	date
eid	fname	lname	age	salary	dept	doj
1	rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
3	kishore	kumar	27	36000	android	2011-10-16
4	abimanyu	biswal	22	NULL	android	2010-02-20
5	soni	kumar	24	21800	.net	2009-06-21
6	anu	_singh	22	12000	db	2010-10-23
7	_dinesh	moh%anty	23	15000	.net	2009-08-26
8	nishala	_kumari	22	18000	db	2008-07-19
1	rajeev	sukla	23	12000	.net	2011-10-23

display employee details in desc order of their first name

```
select * from employees order by fname desc
```



FRS created

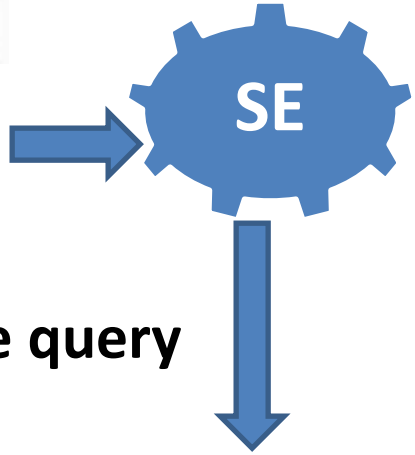
eid	fname	lname	age	salary	dept	doj
2	sowmya	kumari	23	19000	db	2010-11-13
5	soni	kumar	24	21800	.net	2009-06-21
1	rajeev	sukla	23	12000	.net	2011-10-23
1	rajeev	sukla	23	12000	.net	2011-10-23
8	nishala	_kumari	22	18000	db	2008-07-19
3	kishore	kumar	27	36000	android	2011-10-16
6	anu	_singh	22	12000	db	2010-10-23
4	abimanyu	biswal	22	NULL	android	2010-02-20
7	_dinesh	moh%anty	23	15000	.net	2009-08-26

draw the final result set .

employees table

int	varchar(40)	varchar(40)	int	int	varchar(40)	date
eid	fname	lname	age	salary	dept	doj
1	rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
3	kishore	kumar	27	36000	android	2011-10-16
4	abimanyu	biswal	22	NULL	android	2010-02-20
5	soni	kumar	24	21800	.net	2009-06-21
6	anu	_singh	22	12000	db	2010-10-23
7	_dinesh	moh%anty	23	15000	.net	2009-08-26
8	nishala	_kumari	22	18000	db	2008-07-19
1	rajeev	sukla	23	12000	.net	2011-10-23

select * from employees order by lname,fname desc



modifies the query

select * from employees order by lname asc,fname desc

select * from employees order by lname asc,fname desc

int	varchar(40)	varchar(40)	int	int	varchar(40)	date
eid	fname	lname	age	salary	dept	doj
1	rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
3	kishore	kumar	27	36000	android	2011-10-16
4	abimanyu	biswal	22	NULL	android	2010-02-20
5	soni	kumar	24	21800	.net	2009-06-21
6	anu	_singh	22	12000	db	2010-10-23
7	_dinesh	moh%anty	23	15000	.net	2009-08-26
8	nishala	_kumari	22	18000	db	2008-07-19
1	rajeev	sukla	23	12000	.net	2011-10-23

employees table

eid	fname	lname	age	salary	dept	doj
8	nishala	_kumari	22	18000	db	2008-07-19
6	anu	_singh	22	12000	db	2010-10-23
4	abimanyu	biswal	22	NULL	android	2010-02-20
5	soni	kumar	24	21800	.net	2009-06-21
3	kishore	kumar	27	36000	android	2011-10-16
2	sowmya	kumari	23	19000	db	2010-11-13
7	_dinesh	moh%anty	23	15000	.net	2009-08-26
1	rajeev	sukla	23	12000	.net	2011-10-23
1	rajeev	sukla	23	12000	.net	2011-10-23

FRS

order by lab-1

- identify the output for the following query?

select concat(fname,lname) as 'full name', age from patient order by age

- identify the output for the following query?

select fname, lname, bg from patient order by bg desc

note: **ascii** for **+** is **43** and for **–** is **45**

- identify the output for the following query?

select fname, lname, pid from patient order by lname, fname desc

order by lab-2

- identify the result set for the following query?

  select fname,age from patient order by concat('Hello'+fname+lname) desc
- write a query for displaying all patients data in the descending order of their ages?

distinct clause

- distinct clause is used for **eliminating duplicate rows** from result set
- Syntax:

```
select distinct select_list
from table_name
```

int	varchar(40)	varchar(40)	int	int	varchar(40)	date
eid	fname	lname	age	salary	dept	doj
1	rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
3	kishore	kumar	27	36000	android	2011-10-16
4	abimanyu	biswal	22	NULL	android	2010-02-20
5	soni	kumar	24	21800	.net	2009-06-21
6	anu	_singh	22	12000	db	2010-10-23
7	_dinesh	moh%anty	23	15000	.net	2009-08-26
8	nishala	_kumari	22	18000	db	2008-07-19
1	rajeev	sukla	23	12000	.net	2011-10-23

employees table

query: select distinct * from employees

eid	fname	lname	age	salary	dept	doj
1	rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
3	kishore	kumar	27	36000	android	2011-10-16
4	abimanyu	biswal	22	NULL	android	2010-02-20
5	soni	kumar	24	21800	.net	2009-06-21
6	anu	_singh	22	12000	db	2010-10-23
7	_dinesh	moh%anty	23	15000	.net	2009-08-26
8	nishala	_kumari	22	18000	db	2008-07-19
1	rajeev	sukla	23	12000	.net	2011-10-23

SE produces **IRS** without considering distinct clause

INTERMEDIATE RESULT SET

FINAL RESULT SET

eid	fname	lname	age	salary	dept	doj
1	rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
3	kishore	kumar	27	36000	android	2011-10-16
4	abimanyu	biswal	22	NULL	android	2010-02-20
5	soni	kumar	24	21800	.net	2009-06-21
6	anu	_singh	22	12000	db	2010-10-23
7	_dinesh	moh%anty	23	15000	.net	2009-08-26
8	nishala	_kumari	22	18000	db	2008-07-19

select distinct lname from employees

lname
sukla
kumari
kumar
biswal
~~kumar~~
_singh
moh%anty
_kumari
~~sukla~~

SE produces **IRS** without considering distinct clause

INTERMEDIATE RESULT SET

lname
sukla
kumari
Kumar
biswal
_singh
Moh%anty
_kumari

FRS

distinct lab-1

- identify the result set for the following query?

  select distinct fname, age from patient
- identify the result set for the following query?

  select distinct age, bg from patient
- identify the result set for the following query?

  select distinct fname, age,bg from patient order by bg desc

limit

- limit clause is used to fetch **top n rows** or first n rows from a table.
- syntax

  select select_list from table_name limit n

  will fetch first n records from the specified table

  select select_list from table_name limit m,n

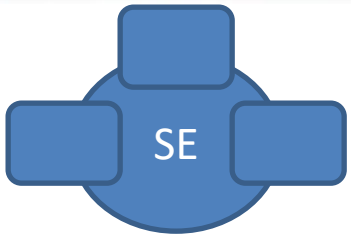
  will fetch n records from the table after first m record

display the first 5 records from employee table

employee

eid	fname	lname	age	salary	dept	doj
1	rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
3	kishore	kumar	27	36000	android	2011-10-16
4	abimanyu	biswal	22	NULL	android	2010-02-20
5	soni	kumar	24	21800	.net	2009-06-21
6	anu	_singh	22	12000	db	2010-10-23
7	_dinesh	moh%anty	23	15000	.net	2009-08-26
8	nishala	_kumari	22	18000	db	2008-07-19
1	rajeev	sukla	23	12000	.net	2011-10-23

select * from employee limit 5



final result set

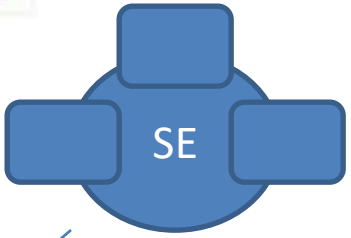
eid	fname	lname	age	salary	dept	doj
1	rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
3	kishore	kumar	27	36000	android	2011-10-16
4	abimanyu	biswal	22	NULL	android	2010-02-20
5	soni	kumar	24	21800	.net	2009-06-21

display fname , lname and age column data for first 3 records from employee table

eid	fname	lname	age	salary	dept	doj
1	rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
3	kishore	kumar	27	36000	android	2011-10-16
4	abimanyu	biswal	22	NULL	android	2010-02-20
5	soni	kumar	24	21800	.net	2009-06-21
6	anu	_singh	22	12000	db	2010-10-23
7	_dinesh	moh%anty	23	15000	.net	2009-08-26
8	nishala	_kumari	22	18000	db	2008-07-19
1	rajeev	sukla	23	12000	.net	2011-10-23

select fname,lname,age from employee limit 3

employee



eid	fname	lname	age	salary	dept	doj
1	rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
3	kishore	kumar	27	36000	android	2011-10-16

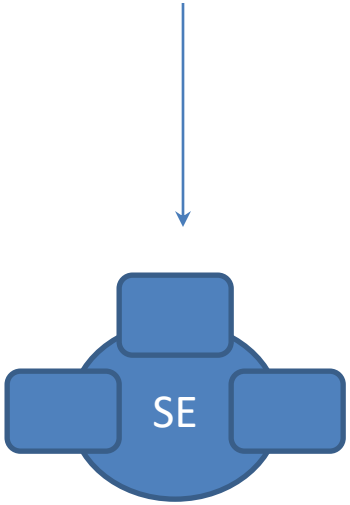
IRS

FRS

fname	lname	age
rajeev	sukla	23
sowmya	kumari	23
kishore	kumar	27

write a query to display the top 3 highest paid employee's fullname and age

```
select concat(fname,' ',lname) as 'fullname' ,age from employee order by salary desc limit 3
```



IRS1

eid	fname	lname	age	salary	dept	doj
3	kishore	kumar	27	36000	android	2011-10-16
5	soni	kumar	24	21800	.net	2009-06-21
2	sowmya	kumari	23	19000	db	2010-11-13
7	nishala	_kumari	22	18000	db	2008-07-19
8	_dinesh	moh%anty	23	15000	.net	2009-08-26
1	rajeev	sukla	23	12000	.net	2011-10-23
6	anu	_singh	22	12000	db	2010-10-23
1	rajeev	sukla	23	12000	.net	2011-10-23
4	abima...	biswal	22	NULL	android	2010-02-20

IRS2

eid	fname	lname	age	salary	dept	doj
3	kishore	kumar	27	36000	android	2011-10-16
5	soni	kumar	24	21800	.net	2009-06-21
2	sowmya	kumari	23	19000	db	2010-11-13

Final Result Set

fullname	age
kishorekumar	27
sonikumar	24
sowmyakumari	23

employee

eid	fname	lname	age	salary	dept	doj
3	kishore	kumar	27	36000	android	2011-10-16
5	soni	kumar	24	21800	.net	2009-06-21
8	_dinesh	moh%anty	23	15000	.net	2009-08-26
1	rajeev	sukla	23	12000	.net	2011-10-23
~~1~~	~~rajeev~~	~~sukla~~	~~23~~	~~12000~~	~~.net~~	~~2011-10-23~~
2	sowmya	kumari	23	19000	db	2010-11-13
4	abimanyu	biswal	22	NULL	android	2010-02-20
6	anu	_singh	22	12000	db	2010-10-23
7	nishala	_kumari	22	18000	db	2008-07-19

select distinct fname,age from employee order by age desc limit 6

fname	age
kishore	27
soni	24
_dinesh	23
rajeev	23
sowmya	23
abimanyu	22

final result set

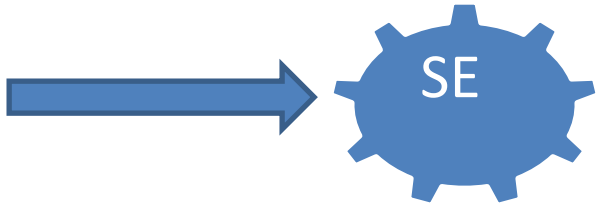
draw the result set for this query

req: display top 2 records from employee table after 5th record

employee

eid	fname	lname	age	salary	dept	doj
1	rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
3	kishore	kumar	27	36000	android	2011-10-16
4	abimanyu	biswal	22	NULL	android	2010-02-20
5	soni	kumar	24	21800	.net	2009-06-21
6	anu	_singh	22	12000	db	2010-10-23
7	_dinesh	moh%anty	23	15000	.net	2009-08-26
8	nishala	_kumari	22	18000	db	2008-07-19
1	rajeev	sukla	23	12000	.net	2011-10-23

query : `select * from employee limit 5,2`



FRS

eid	fname	lname	age	salary	dept	doj
6	anu	_singh	22	12000	db	2010-10-23
7	_dinesh	moh%anty	23	15000	.net	2009-08-26

limit clause lab-1

- identify the output for the following query?

  select fname, lname, age from patient limit 3;
- write a query for displaying top 3 patients details in the descending order of their age?
- ERS:

pid	fname	lname	age	bg
4	hari	kiran	60	b-ve
9	bharat	kumar	56	b-ve
3	madhava	kiran	52	o+ve

where clause

- where clause is used to **filter** the records present in a table.
- where clause can be applied on select or update or delete statements.

syntax

select select_list from table_name where condition

for writing a condition, we have to understand the operators

operators part-1

=,>, >=, <, <= ,!= ,<> , and, or, between, not between ,in, not in, is null, is not null, all, any

conditional and(&&),conditional or(||) are supported in sql.

bitwise and(&),bitwise or(|) are supported in sql but we never use in real time applications

operators part-2

while using **in** or **not in** we must use set of values.

ex: where col1 in (val1,val2,…)

while checking for **null** values we must use either **is null** or **is not null**

while using **between** or **not between** we must use min value first and max value later

req: display employees details whose salary is greater than 20000

eid	fname	lname	age	salary	dept	doj
1	rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
3	kishore	kumar	27	36000	android	2011-10-16
4	abimanyu	biswal	22	NULL	android	2010-02-20
5	soni	kumar	24	21800	.net	2009-06-21
6	anu	_singh	22	12000	db	2010-10-23
7	_dinesh	moh%anty	23	15000	.net	2009-08-26
8	nishala	_kumari	22	18000	db	2008-07-19
1	rajeev	sukla	23	12000	.net	2011-10-23

query: select * from employees where salary>20000

eid	fname	lname	age	salary	dept	doj
3	kishore	kumar	27	36000	android	2011-10-16
5	soni	kumar	24	21800	.net	2009-06-21

req: display all employees details whose salary is between 15000 and 25000

eid	fname	lname	age	salary	dept	doj
1	rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
3	kishore	kumar	27	36000	android	2011-10-16
4	abimanyu	biswal	22	NULL	android	2010-02-20
5	soni	kumar	24	21800	.net	2009-06-21
6	anu	_singh	22	12000	db	2010-10-23
7	_dinesh	moh%anty	23	15000	.net	2009-08-26
8	nishala	_kumari	22	18000	db	2008-07-19
1	rajeev	sukla	23	12000	.net	2011-10-23

query: select * from employees where salary between 15000 and 25000

eid	fname	lname	age	salary	dept	doj
2	sowmya	kumari	23	19000	db	2010-11-13
5	soni	kumar	24	21800	.net	2009-06-21
7	_dinesh	moh%anty	23	15000	.net	2009-08-26
8	nishala	_kumari	22	18000	db	2008-07-19

req: display all employees details whose salary is less than 15000 greater than 25000

eid	fname	lname	age	salary	dept	doj
1	rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
3	kishore	kumar	27	36000	android	2011-10-16
4	abimanyu	biswal	22	NULL	android	2010-02-20
5	soni	kumar	24	21800	.net	2009-06-21
6	anu	_singh	22	12000	db	2010-10-23
7	_dinesh	moh%anty	23	15000	.net	2009-08-26
8	nishala	_kumari	22	18000	db	2008-07-19
1	rajeev	sukla	23	12000	.net	2011-10-23

query: select * from employees where salary not between 15000 and 25000

eid	fname	lname	age	salary	dept	doj
1	rajeev	sukla	23	12000	.net	2011-10-23
3	kishore	kumar	27	36000	android	2011-10-16
6	anu	_singh	22	12000	db	2010-10-23
1	rajeev	sukla	23	12000	.net	2011-10-23

req: display all employees details whose salary is equal to 15000 or21800 or 36000

eid	fname	lname	age	salary	dept	doj
1	rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
3	kishore	kumar	27	36000	android	2011-10-16
4	abimanyu	biswal	22	NULL	android	2010-02-20
5	soni	kumar	24	21800	.net	2009-06-21
6	anu	_singh	22	12000	db	2010-10-23
7	_dinesh	moh%anty	23	15000	.net	2009-08-26
8	nishala	_kumari	22	18000	db	2008-07-19
1	rajeev	sukla	23	12000	.net	2011-10-23

query: select * from employees where salary in(15000,21800,36000)

eid	fname	lname	age	salary	dept	doj
3	kishore	kumar	27	36000	android	2011-10-16
5	soni	kumar	24	21800	.net	2009-06-21
7	_dinesh	moh%anty	23	15000	.net	2009-08-26

req: display all employees details whose salary is not equal to 15000 or21800 or 36000

eid	fname	lname	age	salary	dept	doj
1	rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
3	kishore	kumar	27	36000	android	2011-10-16
4	abimanyu	biswal	22	NULL	android	2010-02-20
5	soni	kumar	24	21800	.net	2009-06-21
6	anu	_singh	22	12000	db	2010-10-23
7	_dinesh	moh%anty	23	15000	.net	2009-08-26
8	nishala	_kumari	22	18000	db	2008-07-19
1	rajeev	sukla	23	12000	.net	2011-10-23

query: select * from employees where salary not in(15000,21800,36000)

eid	fname	lname	age	salary	dept	doj
1	rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
6	anu	_singh	22	12000	db	2010-10-23
8	nishala	_kumari	22	18000	db	2008-07-19
1	rajeev	sukla	23	12000	.net	2011-10-23

req: display all employees details whose salary is null

eid	fname	lname	age	salary	dept	doj
1	rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
3	kishore	kumar	27	36000	android	2011-10-16
4	abimanyu	biswal	22	NULL	android	2010-02-20
5	soni	kumar	24	21800	.net	2009-06-21
6	anu	_singh	22	12000	db	2010-10-23
7	_dinesh	moh%anty	23	15000	.net	2009-08-26
8	nishala	_kumari	22	18000	db	2008-07-19
1	rajeev	sukla	23	12000	.net	2011-10-23

query: select * from employees where salary is null

eid	fname	lname	age	salary	dept	doj
4	abimanyu	biswal	22	NULL	android	2010-02-20

req: display all employees details whose salary is not null

eid	fname	lname	age	salary	dept	doj
1	rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
3	kishore	kumar	27	36000	android	2011-10-16
4	abimanyu	biswal	22	NULL	android	2010-02-20
5	soni	kumar	24	21800	.net	2009-06-21
6	anu	_singh	22	12000	db	2010-10-23
7	_dinesh	moh%anty	23	15000	.net	2009-08-26
8	nishala	_kumari	22	18000	db	2008-07-19
1	rajeev	sukla	23	12000	.net	2011-10-23

query: select * from employees where salary is not null

eid	fname	lname	age	salary	dept	doj
1	rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
3	kishore	kumar	27	36000	android	2011-10-16
5	soni	kumar	24	21800	.net	2009-06-21
6	anu	_singh	22	12000	db	2010-10-23
7	_dinesh	moh%anty	23	15000	.net	2009-08-26
8	nishala	_kumari	22	18000	db	2008-07-19
1	rajeev	sukla	23	12000	.net	2011-10-23

where clause lab-1

- write a query for displaying all patients details whose age is greater than 45?

  select * from patient where age>45
- write a query for displaying all patients details whose age is between 40 and 50 (write query in all possible ways)?

  select * from patient where age between 40 and 50

  select * from patient where age>=40 and age<=50
- write a query for displaying all patients details whose age is greater than 40 and whose bg is not o+ve?

  select * from patient where age>40 and bg!='o+ve'

where clause lab-2

- write a query for displaying all patient details whose bg is not null?

  select * from patient where bg is not null

- write a query for displaying all patient details whose age is equal to 42 or 36 or 60 (write query using all possible ways)?

  select * from patient where age=42 or age=36 or age=60

  select * from patient where age in (42,36,60)

- write a query for displaying all patient details whose age is not equal to 42 and 36 and 60 (write query using all possible ways)?

  select * from patient where age!=42 or age!=36 or age!=60

  select * from patient where age not in (42,36,60)

where clause lab-3

- identify the result set for the following query?

  select * from patient where age>40 order by bg desc limit 3

  (student must identify which portion of the query is executed first, also write IRS and FRS)

like clause & pattern matching

- using like clause we can find data which is **matching** to a **specific pattern**.

  syntax :

  where column|expression|variable like 'pattern'

- wildcard characters used in pattern matching

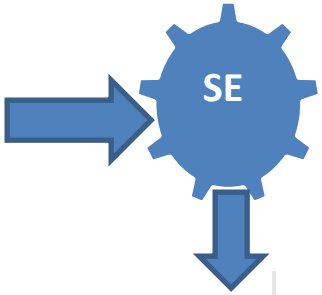
wildcard character	description
%	0 or more characters
_	any single character

employee

eid	fname	lname	age	salary	dept	doj
1	rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
3	kishore	kumar	27	36000	android	2011-10-16
4	abimanyu	biswal	22	NULL	android	2010-02-20
5	soni	kumar	24	21800	.net	2009-06-21
6	anu	_singh	22	12000	db	2010-10-23
7	_dinesh	moh%anty	23	15000	.net	2009-08-26
8	nishala	_kumari	22	18000	db	2008-07-19
1	rajeev	sukla	23	12000	.net	2011-10-23

requirement: display employee details who is having fname starting with character 'a'

query: select * from employee where fname like 'a%'



FRS

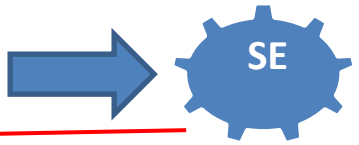
eid	fname	lname	age	salary	dept	doj
4	abimanyu	biswal	22	NULL	android	2010-02-20 00:00:00.000
6	anu	_singh	22	12000	db	2010-10-23 00:00:00.000

employee

eid	fname	lname	age	salary	dept	doj
1	rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
3	kishore	kumar	27	36000	android	2011-10-16
4	abimanyu	biswal	22	NULL	android	2010-02-20
5	soni	kumar	24	21800	.net	2009-06-21
6	anu	_singh	22	12000	db	2010-10-23
7	_dinesh	moh%anty	23	15000	.net	2009-08-26
8	nishala	_kumari	22	18000	db	2008-07-19
1	rajeev	sukla	23	12000	.net	2011-10-23

requirement: display employee details who is having fname ending with character 'a'

query: select * from employee where fname like '%a'



FRS

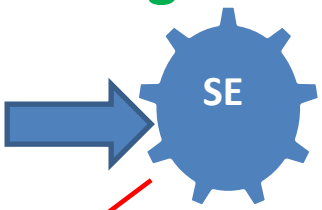
eid	fname	lname	age	salary	dept	doj
2	sowmya	kumari	23	19000	db	2010-11-13 00:00:00.000
8	nishala	_kumari	22	18000	.db	2008-07-19 00:00:00.000

employee

eid	fname	lname	age	salary	dept	doj
1	rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
3	kishore	kumar	27	36000	android	2011-10-16
4	abimanyu	biswal	22	NULL	android	2010-02-20
5	soni	kumar	24	21800	.net	2009-06-21
6	anu	_singh	22	12000	db	2010-10-23
7	_dinesh	moh%anty	23	15000	.net	2009-08-26
8	nishala	_kumari	22	18000	db	2008-07-19
1	rajeev	sukla	23	12000	.net	2011-10-23

requirement: display employee details whose fname is having character 'a' in second position

query: select * from employee where fname like '_a%'



FRS

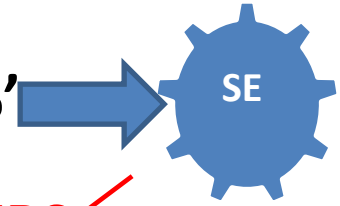
eid	fname	lname	age	salary	dept	doj
1	rajeev	sukla	23	12000	.net	2011-10-23 00:00:00.000
1	rajeev	sukla	23	12000	.net	2011-10-23 00:00:00.000

employees

eid	fname	lname	age	salary	dept	doj
1	rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
3	kishore	kumar	27	36000	android	2011-10-16
4	abimanyu	biswal	22	NULL	android	2010-02-20
5	soni	kumar	24	21800	.net	2009-06-21
6	anu	_singh	22	12000	db	2010-10-23
7	_dinesh	moh%anty	23	15000	.net	2009-08-26
8	nishala	_kumari	22	18000	db	2008-07-19
1	rajeev	sukla	23	12000	.net	2011-10-23

requirement: display employee details whose fname is having character 'a' anywhere

query: select * from employees where fname like '%a%'



FRS

eid	fname	lname	age	salary	dept	doj
1	rajeev	sukla	23	12000	.net	2011-10-23 00:00:00.000
2	sowmya	kumari	23	19000	db	2010-11-13 00:00:00.000
4	abimanyu	biswal	22	NULL	android	2010-02-20 00:00:00.000
6	anu	_singh	22	12000	db	2010-10-23 00:00:00.000
8	nishala	_kumari	22	18000	.db	2008-07-19 00:00:00.000
1	rajeev	sukla	23	12000	.net	2011-10-23 00:00:00.000

employees

eid	fname	lname	age	salary	dept	doj
1	rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
3	kishore	kumar	27	36000	android	2011-10-16
4	abimanyu	biswal	22	NULL	android	2010-02-20
5	soni	kumar	24	21800	.net	2009-06-21
6	anu	_singh	22	12000	db	2010-10-23
7	_dinesh	moh%anty	23	15000	.net	2009-08-26
8	nishala	_kumari	22	18000	db	2008-07-19
1	rajeev	sukla	23	12000	.net	2011-10-23

requirement: display employee details whose lname having character 'a' in second last position

query: select * from employees where lname like '%a_'

FRS

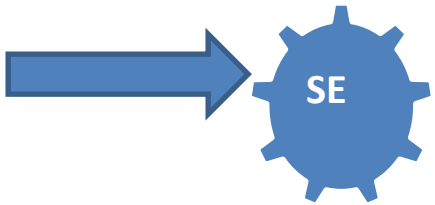
eid	fname	lname	age	salary	dept	doj
3	kishore	kumar	27	36000	android	2011-10-16
4	abimanyu	biswal	22	NULL	android	2010-02-20
5	soni	kumar	24	21800	.net	2009-06-21

employees

eid	fname	lname	age	salary	dept	doj
1	rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
3	kishore	kumar	27	36000	android	2011-10-16
4	abimanyu	biswal	22	NULL	android	2010-02-20
5	soni	kumar	24	21800	.net	2009-06-21
6	anu	_singh	22	12000	db	2010-10-23
7	_dinesh	moh%anty	23	15000	.net	2009-08-26
8	nishala	_kumari	22	18000	db	2008-07-19
1	rajeev	sukla	23	12000	.net	2011-10-23

requirement: display employee details whose fname having character 'a' anywhere in fname and data must displayed in descending order of their salaries

query: select * from employees where fname like '%a%' order by salary desc



employees

eid	fname	lname	age	salary	dept	doj
1	rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
3	kishore	kumar	27	36000	android	2011-10-16
4	abimanyu	biswal	22	NULL	android	2010-02-20
5	soni	kumar	24	21800	.net	2009-06-21
6	anu	_singh	22	12000	db	2010-10-23
7	_dinesh	moh%anty	23	15000	.net	2009-08-26
8	nishala	_kumari	22	18000	db	2008-07-19
1	rajeev	sukla	23	12000	.net	2011-10-23

FRS created

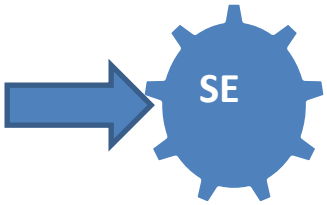
eid	fname	lname	age	salary	dept	doj
2	sowmya	kumari	23	19000	db	2010-11-13
8	nishala	_kumari	22	18000	db	2008-07-19
1	rajeev	sukla	23	12000	.net	2011-10-23
1	rajeev	sukla	23	12000	.net	2011-10-23
6	anu	_singh	22	12000	db	2010-10-23
4	abimanyu	biswal	22	NULL	android	2010-02-20

employees

eid	fname	lname	age	salary	dept	doj
1	rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
3	kishore	kumar	27	36000	android	2011-10-16
4	abimanyu	biswal	22	NULL	android	2010-02-20
5	soni	kumar	24	21800	.net	2009-06-21
6	anu	_singh	22	12000	db	2010-10-23
7	_dinesh	moh%anty	23	15000	.net	2009-08-26
8	nishala	_kumari	22	18000	db	2008-07-19
1	rajeev	sukla	23	12000	.net	2011-10-23

requirement: display employee details whose fname are having character 'i' in third position

query: **select * from employees where fname like '__i%'**



FRS created

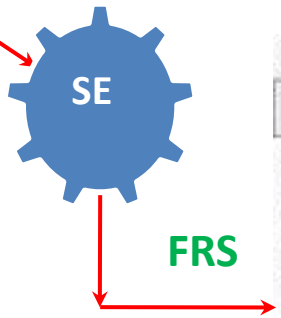
eid	fname	lname	age	salary	dept	doj
4	abimanyu	biswal	22	NULL	android	2010-02-20
7	_dinesh	moh%anty	23	15000	.net	2009-08-26

employees

eid	fname	lname	age	salary	dept	doj
1	rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
3	kishore	kumar	27	36000	android	2011-10-16
4	abimanyu	biswal	22	NULL	android	2010-02-20
5	soni	kumar	24	21800	.net	2009-06-21
6	anu	_singh	22	12000	db	2010-10-23
7	_dinesh	moh%anty	23	15000	.net	2009-08-26
8	nishala	_kumari	22	18000	db	2008-07-19
1	rajeev	sukla	23	12000	.net	2011-10-23

requirement: display employee details whose fname are not ending with character 'a'

query: select * from employees where fname not like '%a'



eid	fname	lname	age	salary	dept	doj
1	rajeev	sukla	23	12000	.net	2011-10-23
3	kishore	kumar	27	36000	android	2011-10-16
4	abimanyu	biswal	22	NULL	android	2010-02-20
5	soni	kumar	24	21800	.net	2009-06-21
6	anu	_singh	22	12000	db	2010-10-23
7	_dinesh	moh%anty	23	15000	.net	2009-08-26
1	rajeev	sukla	23	12000	.net	2011-10-23

like clause lab-1

- write a query for displaying all patient details whose fname's are starting with character **m**?
- write a query for displaying all patients full names whose lnames are ending with **i?**
- write a query for displaying all patients details whose lnames are having character **a** in the second position from ending ?

  (ex : kir**a**n is having character **a** in the **second position** from **ending**)

like clause lab-2

- write a query for displaying all patient details whose fnames are having character **r** in the 3rd position from ending?
- write a query for displaying all patient details whose fnames are having character **i** any where ?

GENERIC INSERT,UPDATE AND DELETE STATEMENTS

USED TO MODIFY OUR TABLE DATA

syntax for generic insert statement

insert into table_name(c1,c2,c3….)

values(v1,v2,v3…)

note: insert statement will **not** contain **where** clause

syntax for generic update statement

SET SQL_SAFE_UPDATES = 0; //remove security

update table_name set c1=v1,c2=v2, ….[where condition]

note: update statement without where will update the complete table data

syntax for generic delete statement

delete from table_name [where condition]

note: delete statement without where will delete all data present in a table.

req: insert a new record into employee table with eid=9, fname=giri, lname=babu, age=27, salary 18000, dept=.net, doj=06-24-2014

query : insert into employees(eid,fname,lname,age,salary,dept,doj) values(9,'giri','babu',27,18000,'.net','2014-06-24')

eid	fname	lname	age	salary	dept	doj
1	Rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
3	Kishore	kumar	27	36000	android	2011-10-16
4	Abimanyu	biswal	22	NULL	android	2010-02-20
5	sony	Kumar	24	21800	.net	2009-06-21
6	anu	_singh	22	12000	db	2010-10-23
7	_dinesh	moh%anty	23	15000	.net	2009-08-19
8	nishala	_kumari	22	18000	db	2008-07-19
1	Rajeev	sukla	23	12000	.net	2011-10-23

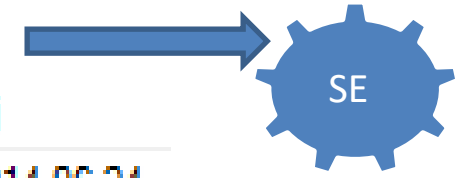
employees

eid	fname	lname	age	salary	dept	doj
9	giri	babu	27	18000	.net	2014-06-24
1	Rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
3	Kishore	kumar	27	36000	android	2011-10-16
4	Abimanyu	biswal	22	NULL	android	2010-02-20
5	sony	Kumar	24	21800	.net	2009-06-21
6	anu	_singh	22	12000	db	2010-10-23
7	_dinesh	moh%anty	23	15000	.net	2009-08-19
8	nishala	_kumari	22	18000	db	2008-07-19
1	Rajeev	sukla	23	12000	.net	2011-10-23

employees

req: update employee table with fname to rice, lname to paul whose eid is 9

query: update employees set fname='rice', lname='paul' where eid=9



eid	fname	lname	age	salary	dept	doj
9	rice	paul	27	18000	.net	2014-06-24
1	Rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
3	Kishore	kumar	27	36000	android	2011-10-16
4	Abimanyu	biswal	22	NULL	android	2010-02-20
5	sony	Kumar	24	21800	.net	2009-06-21
6	anu	_singh	22	12000	db	2010-10-23
7	_dinesh	moh%anty	23	15000	.net	2009-08-19
8	nishala	_kumari	22	18000	db	2008-07-19
1	Rajeev	sukla	23	12000	.net	2011-10-23

employees employees

req: delete a record from employee table whose eid is 9

query : delete from employees where eid=9 ;

eid	fname	lname	age	salary	dept	doj
9	rice	paul	27	18000	.net	2014-06-24
1	Rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
3	Kishore	kumar	27	36000	android	2011-10-16
4	Abimanyu	biswal	22	NULL	android	2010-02-20
5	sony	Kumar	24	21800	.net	2009-06-21
6	anu	_singh	22	12000	db	2010-10-23
7	_dinesh	moh%anty	23	15000	.net	2009-08-19
8	nishala	_kumari	22	18000	db	2008-07-19
1	Rajeev	sukla	23	12000	.net	2011-10-23

employees

eid	fname	lname	age	salary	dept	doj
1	Rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
3	Kishore	kumar	27	36000	android	2011-10-16
4	Abimanyu	biswal	22	NULL	android	2010-02-20
5	sony	Kumar	24	21800	.net	2009-06-21
6	anu	_singh	22	12000	db	2010-10-23
7	_dinesh	moh%anty	23	15000	.net	2009-08-19
8	nishala	_kumari	22	18000	db	2008-07-19
1	Rajeev	sukla	23	12000	.net	2011-10-23

employees

insert update delete lab-1

- write a query for inserting the following patient details into patient table?

  patient id=10, fname='ahana' , lname='kumar' , age=78 , bg='o+ve' (write query using all possible ways)
- write a query for inserting the following patient details into patient table?

  patient id=11, fname='silli', lname='suresh' , age=81, bg is null (write query using all possible ways)

insert update delete lab-2

- write a query for updating the 10th patient (pid=10) with the following details?

  fname='raja' lname='raveender' age=66 bg='o-ve'
- write a query for updating the 11th patient (pid=11) with the following details?

  fname='meena' lname='kumari'
- write a query for deleting 10th and 11th patients from patient table?

delete

drop

truncate

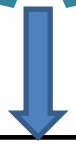
delete without where

- a delete statement without where clause will delete complete table data.
- delete statement is a **logged operation**.
- delete operation is a **reversible operation**.
- delete operation is **slow** compared to truncate operation.

patient table

pid	fname	lname	age	bg
1	Madhava	reddy	45	O+ve
2	Hari	kiran	60	B-ve

delete from patient



similar table will be created in .ldf

.ldf

pid	fname	lname	age	bg
1	Madhava	reddy	45	O+ve
2	Hari	kiran	60	B-ve

truncate table

truncate table statement will delete all data present in a table.

truncate statement is **not** a logged operation and hence we **can't rollback** this operation.

syntax: truncate table table_name

"truncate" operation is faster than "delete without where clause".

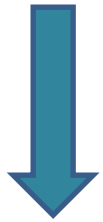
truncate statement **can't contain where** clause

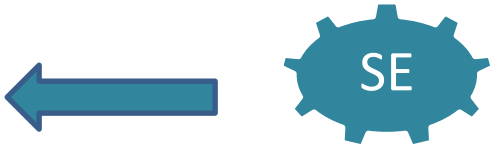
patient table

pid	fname	lname	age	bg
1	Madhava	Reddy	45	O+ve
4	Hari	kiran	60	B-ve
3	Madhava	kiran	52	O+ve
5	Veena	kumari	42	Null
6	K_iran	kumar	39	B-ve
2	Abhinav	bandra	45	O-ve
7	Mahes%h	Nambotri	36	B+ve

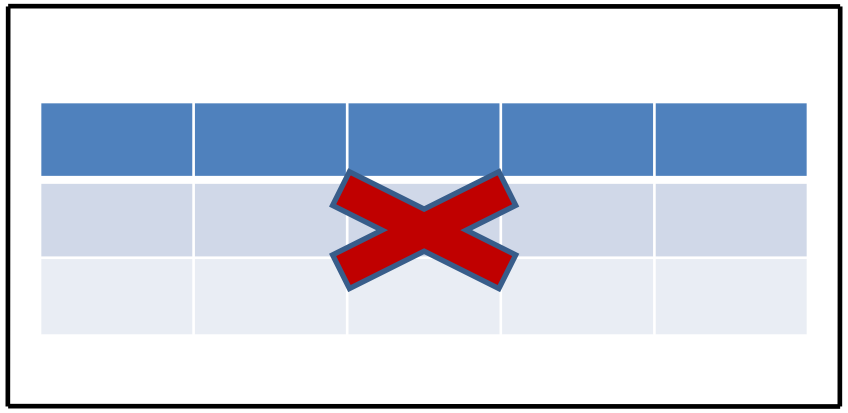
table structure will not delete

truncate table patient





.ldf



no similar table will be created in .ldf

drop table

- drop table operation will delete the table from the database and also the related constraints and indexes.
- **syntax:**

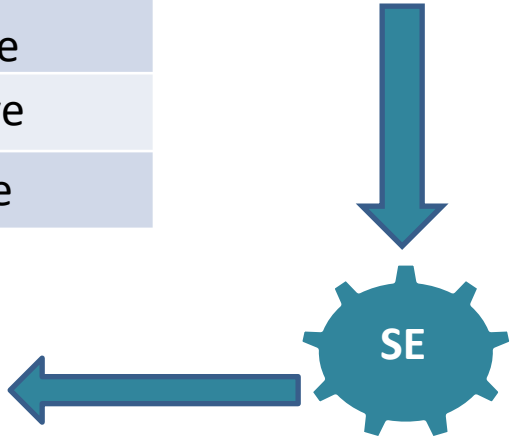
  drop table table_name
- drop is not a logged operation

patient table

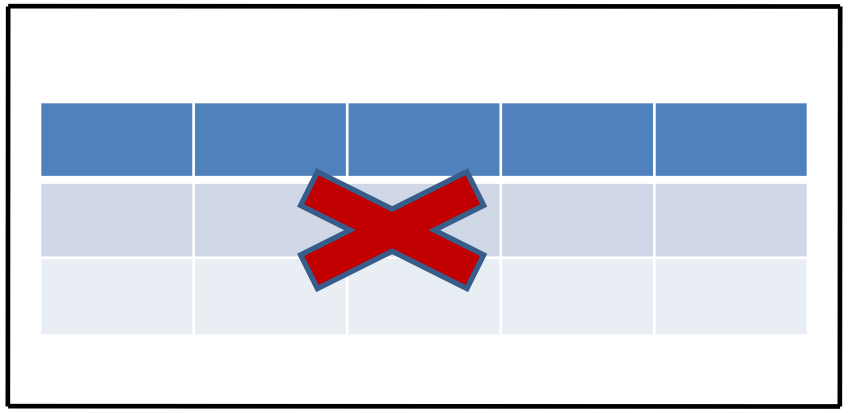
pid	fname	lname	age	bg
1	Madhava	Reddy	45	O+ve
4	Hari	kiran	60	B-ve
3	Madhava	kiran	52	O+ve
5	Veena	kumari	42	Null
6	K_iran	kumar	39	B-ve
2	Abhinav	bandra	45	O-ve
7	Mahes%h	Nambotri	36	B+ve

table structure also deleted

drop table patient

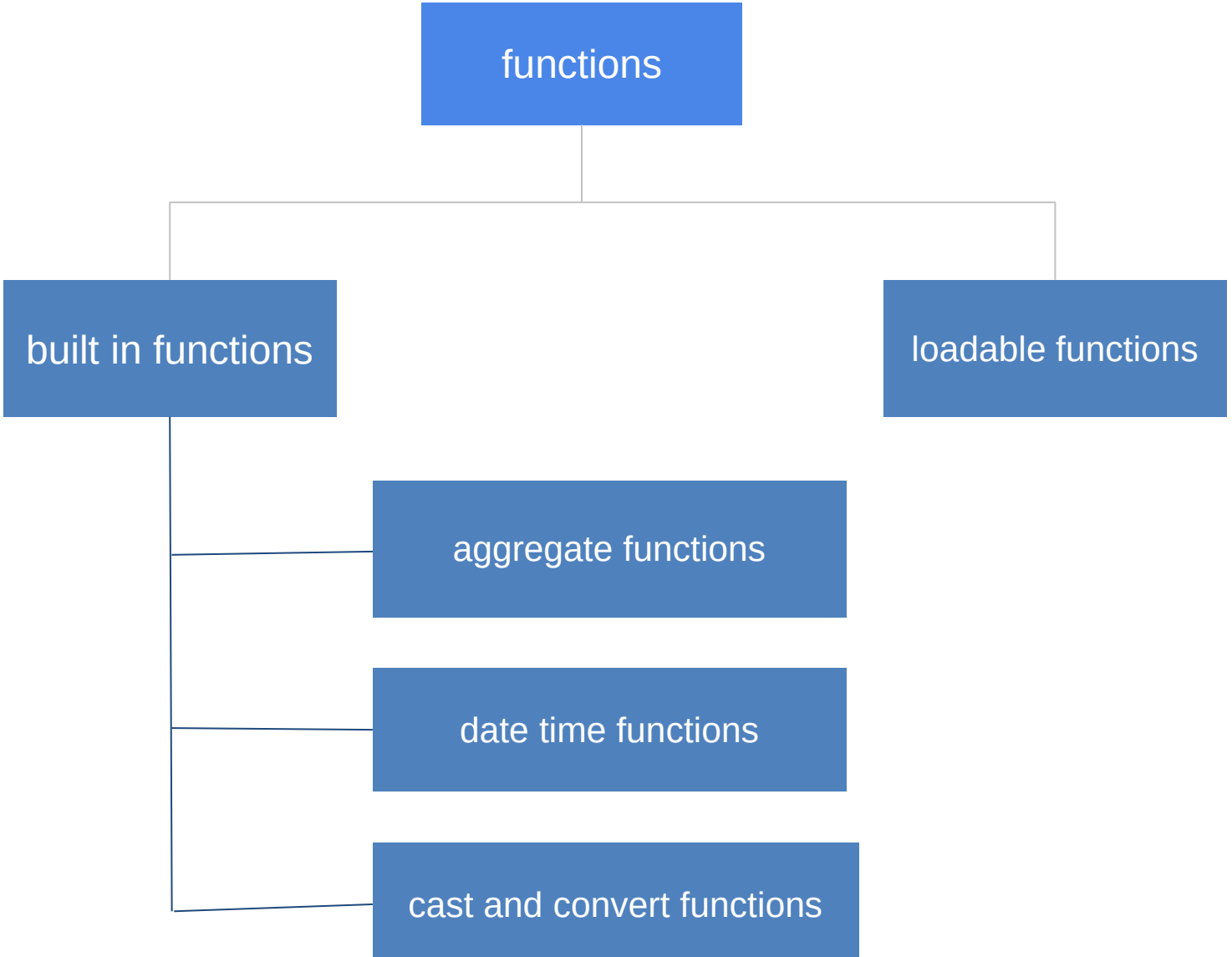


.ldf

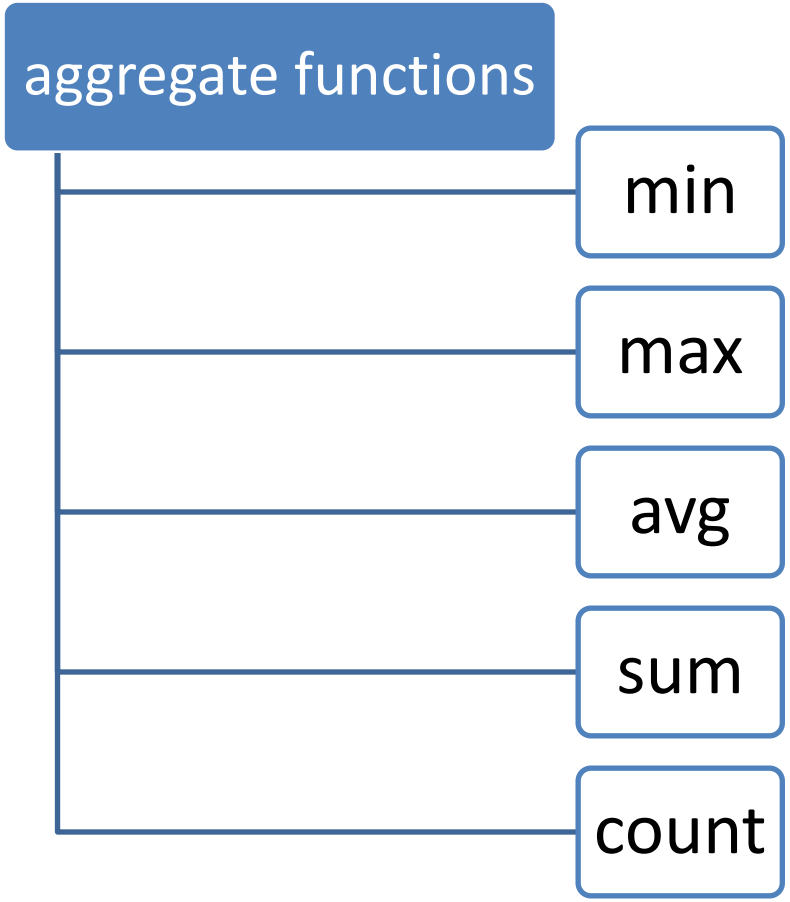


no similar table will be created in .ldf

functions


functions
built in functions
loadable functions
aggregate functions
date time functions
cast and convert functions

aggregate functions



note2: if we use **columns associated** with **aggregate functions** and columns **without** any aggregate function **together** in **select list**, it may give **wrong** output.

note 1: we can't **use** aggregate **functions** in **where** clause

aggregate function sample 1

eid	fname	lname	age	salary	dept	doj
1	rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
3	kishore	kumar	27	36000	android	2011-10-16
4	abimanyu	biswal	22	NULL	android	2010-02-20
5	soni	kumar	24	21800	.net	2009-06-21
6	anu	_singh	22	12000	db	2010-10-23
7	_dinesh	moh%anty	23	15000	.net	2009-08-26
8	nishala	_kumari	22	18000	db	2008-07-19
1	rajeev	sukla	23	12000	.net	2011-10-23

employee

tot_salary
145800

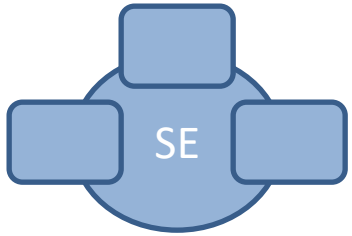
avg(age)
23.2222

select avg(age) from employee

select min(salary) as 'min_sal' from employee

select max(salary) as 'max_sal' from employee

select sum(salary) as 'tot_salary' from employee



sample 2

employee

eid	fname	lname	age	salary	dept	doj
1	rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
3	kishore	kumar	27	36000	android	2011-10-16
4	abimanyu	biswal	22	NULL	android	2010-02-20
5	soni	kumar	24	21800	.net	2009-06-21
6	anu	_singh	22	12000	db	2010-10-23
7	_dinesh	moh%anty	23	15000	.net	2009-08-26
8	nishala	_kumari	22	18000	db	2008-07-19
1	rajeev	sukla	23	12000	.net	2011-10-23

average salary
18225.0000

select avg(salary) as 'average salary' from employee

note : average = sum / count.
SE will **not** count the **null** values

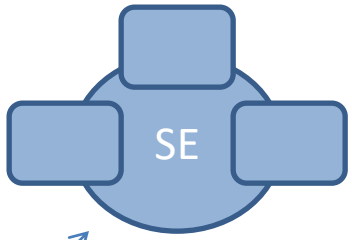
sum = 145800
count (salary) = 8

count () sample

employee

eid	fname	lname	age	salary	dept	doj
1	rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
3	kishore	kumar	27	36000	android	2011-10-16
4	abimanyu	biswal	22	NULL	android	2010-02-20
5	soni	kumar	24	21800	.net	2009-06-21
6	anu	_singh	22	12000	db	2010-10-23
7	_dinesh	moh%anty	23	15000	.net	2009-08-26
8	nishala	_kumari	22	18000	db	2008-07-19
1	rajeev	sukla	23	12000	.net	2011-10-23

re	result
9	4



select count(*) as ‘result’ from employee

select count(*) as ‘result’ from employee where dept=‘.net’

aggregate functions lab-1

- write a query/program for displaying youngest patient age, eldest patient age, sum of all patients age and average age of patients?

ERS:

min age	max age	tot age	avg age
36	60	466	46

- identify the output for the following query?

query: select fname from patient where min(age)=age;

aggregate functions lab-2

- write a program for displaying youngest and eldest patients fnames?

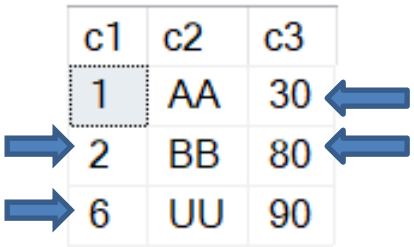
ERS:

youngest	eldest
mahes%h	hari

count function

- count function gives total number of matched records count from an intermediate result set.

```
select COUNT(*) as 'matching rows'  from T1 where c1>1 and c3<90
```



c1	c2	c3
1	AA	30
2	BB	80
6	UU	90

T1

matching rows
1

FRS

c1	c2	c3
2	BB	80

IRS

```
select COUNT(c2) as 'matching rows'  from T1 where c1>1 and c3<90
```

matching rows
1

FRS

c2
BB

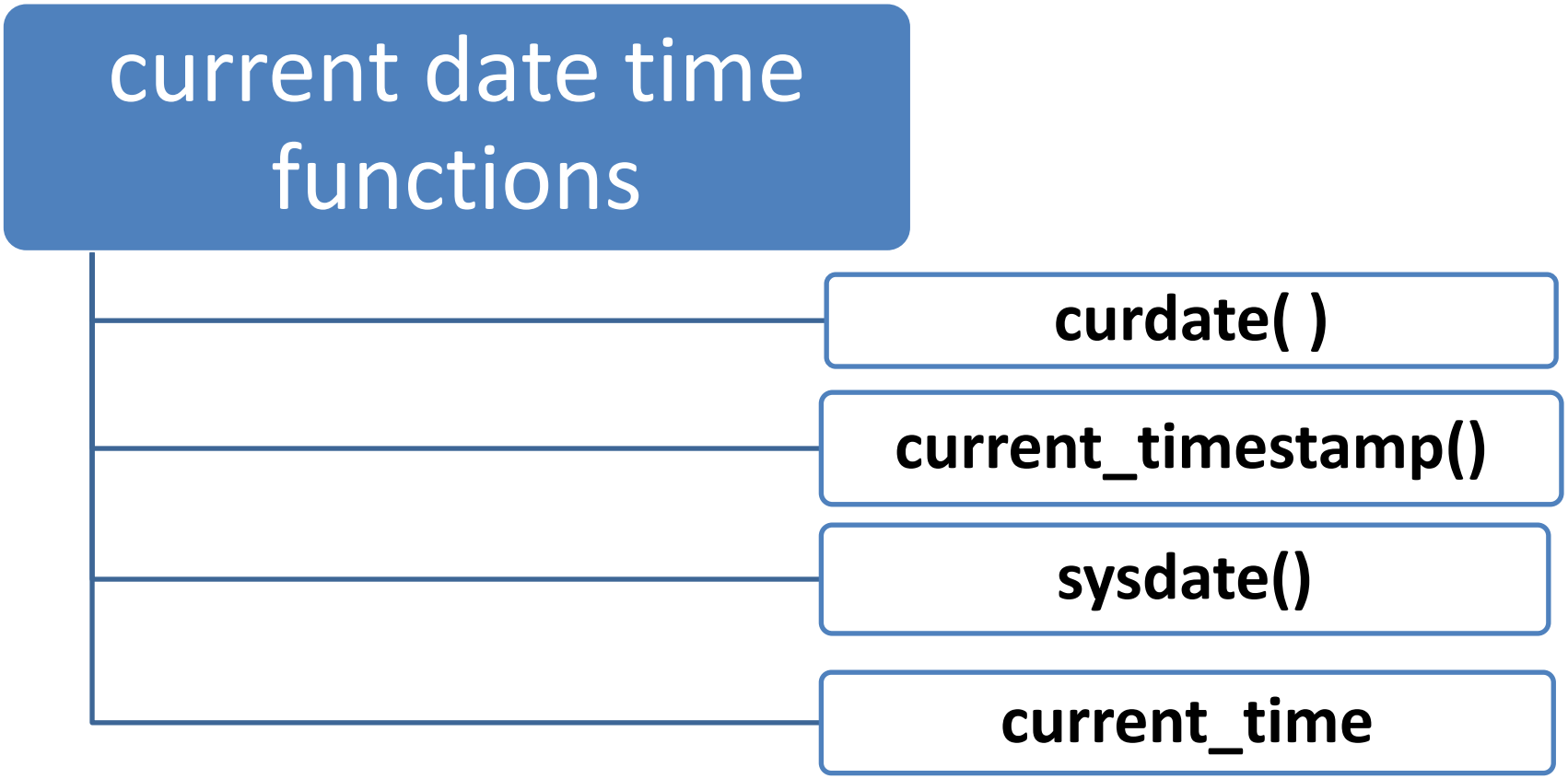
IRS

count function lab-1

- identify the output for the following query?

```
select COUNT(age) as 'tot rows',MAX(Age) as 'max age' from patient where age>45
```

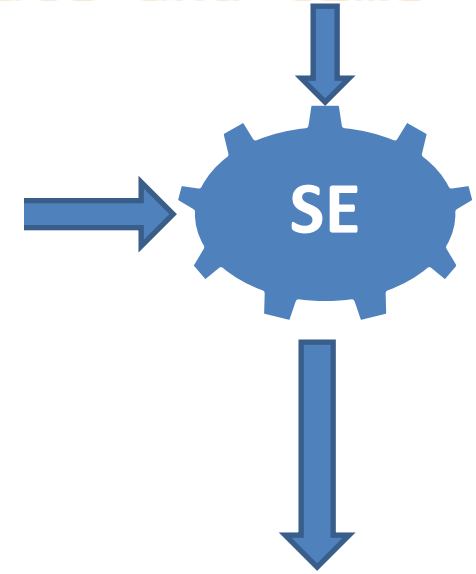
date time functions



current date time functions sample

```
select current_timestamp() as'current date and time'
```

```
select curselect current_time'
```



current date and time
2022-02-28 12:02:48

functions which gives date time parts

- day() → will return day number from a given date or date time value
- month() → will return month number from a given date or date time value
- year() → will return year number from a given date or date time value
- hour() → will return hour number from a given time or date time value
- minute() → will return minute number from a given time or date time value
- dayname()→will return day name from a given date or date time value
- monthname()→will return month name from a given date or date time value

samples 1

```
select year('2021-09-09')
```

year('2021-09-09')
2021

```
select month(curdate())
```

month(curdate())
2

```
select day(sysdate())
```

day(sysdate())
28

sample 2

```
select hour(sysdate())
```

hour(sysdate())
14

```
select minute(sysdate())
```

minute(sysdate())
21

```
select dayname(curdate())
```

dayname(curdate())
Monday

```
select monthname(curdate())
```

monthname(curdate())
February

date _add () function

date_add() function is used to **add** a specified **time** or **date** interval to a specified date and then return the date.

syntax:

date_add(date, interval value datepart)

note : datepart can be second, minute, hour, day, year, month, etc.

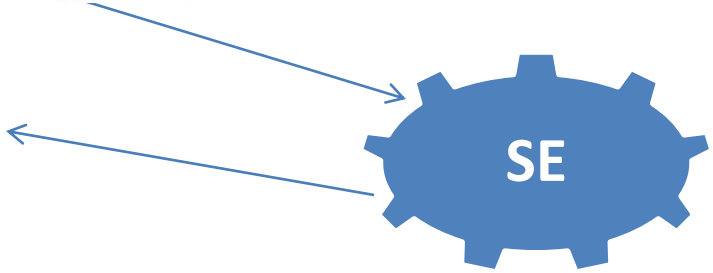
example:

req : add 20 years to the current system date and display it

query :

```
select curdate() as 'current date',
date_add(curdate(),interval 20 year) as 'new date'
```

current date	new date
2022-02-28	2042-02-28



date_sub() function

date_sub() function **subtracts** a **time/date interval** from a date and then returns it.

syntax :

date_sub(date, interval value datepart)

note : datepart can be second, minute, hour, day, year, month, etc.

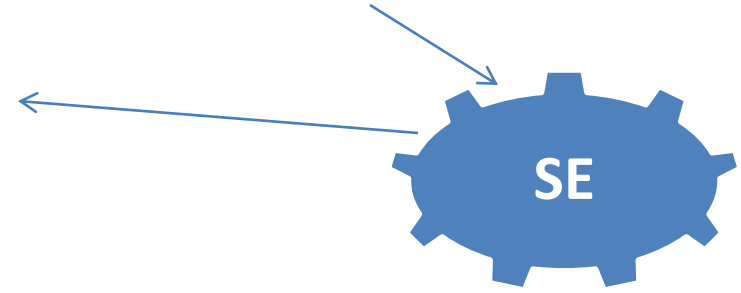
example:

req: subtract 15 months from the date '2020-01-01' and display the result

query:

```
select date_sub('2020-01-01', interval 15 month)
```

date_sub('2020-01-01', interval 15 month)
2018-10-01



datediff() function

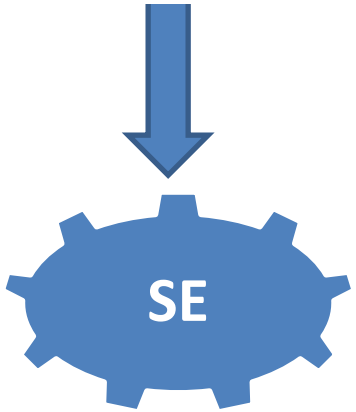
datediff() function returns the **number of days** between two dates

syntax :

datediff(date1, date2)

example :

```
select datediff(curdate(),'2020-12-31')
```



datediff(curdate(),'2020-12-31')
424

date time functions lab-1

- write a program for adding 8 years 3 months and10 days to current date and display the resultant date?

cast and convert functions

both cast() and convert() **converts** a value of any type into the **specified** datatype.

syntax :

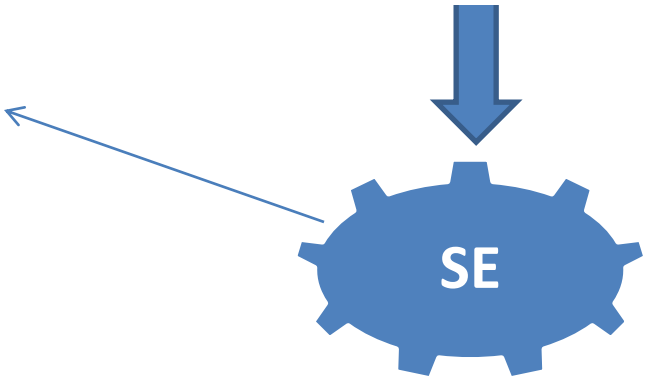
cast(value as datatype)
convert(value, datatype)

here the value is the value to convert and datatype can be any of the following :
date, datetime, time, char, unsigned, signed, binary

example :

```
selselect convert('20201023123456',datetime )
```

convert('20201023123456',datetime)
2020-10-23 12:34:56



group by and having clauses

- using group by clause we can group data present in a table **based on** one or more columns.

syntax :

```
select  column_names_used_in_group_by_clause
or columns_linked_with_aggregate_functions
from table name
group by column1,column2,.....
```

note :

all columns specified **in** select list must be present in group by clause

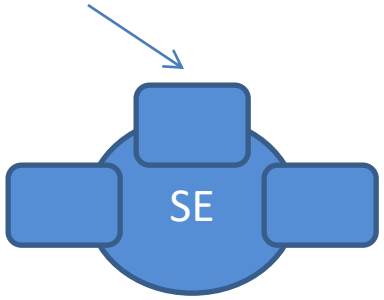
select c1,c2 from t1 group by c1,c2 ✓

select c1,c2 from t1 group by c1,c2,c3 ✓

select c1,c2,avg(c3) from t1 group by c1,c2 ✓

group by internals

select bg from patients group by bg



FRS

bg
o+ve
o-ve
b-ve
b+ve

patient

P_id	Fname	Lname	age	bg
1	Madhava	Reddy	45	o+ve
4	Mahadeva	kiran	52	o+ve

P_id	Fname	Lname	age	bg
2	Abhinav	bandra	45	o-ve
2	Abhinav	bandra	45	o-ve

P_id	Fname	Lname	age	bg
3	Hari	kiran	60	b-ve
6	K_iran	kumar	39	b-ve
8	Rahul	kumar	46	b-ve
9	Bharat	kumar	56	b-ve

P_id	Fname	Lname	age	bg
7	Mahes%h	nambotri	36	b+ve

P_id	Fname	Lname	age	bg
1	Madhava	Reddy	45	o+ve
2	Abhinav	bandra	45	o-ve
3	Hari	kiran	60	b-ve
4	Mahadeva	kiran	52	o+ve
6	K_iran	kumar	39	b-ve
2	Abhinav	bandra	45	o-ve
7	Mahes%h	nambotri	36	b+ve
8	Rahul	kumar	46	b-ve
9	Bharat	kumar	56	b-ve

having clause

having clause is used to filter rows which are produced by group by clause

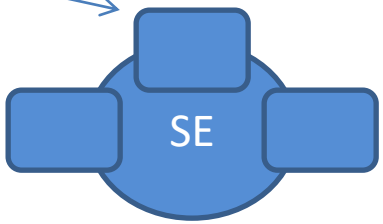
we can't write having clause without group by clause

having clause sample

select bg from patient group by bg having avg(age)>45

FRS

bg
o+ve
b-ve



P_id	Fname	Lname	age	bg
1	Madhava	Reddy	45	o+ve
4	Mahadeva	kiran	52	o+ve

✓ avg(age)>45

P_id	Fname	Lname	age	bg
2	Abhinav	bandra	45	o-ve
2	Abhinav	bandra	45	o-ve

✗ avg(age)>45

P_id	Fname	Lname	age	bg
3	Hari	kiran	60	b-ve
6	K_iran	kumar	39	b-ve
8	Rahul	kumar	46	b-ve
9	Bharat	kumar	56	b-ve

✓ avg(age)>45

P_id	Fname	Lname	age	bg
7	Mahes%h	nambotri	36	b+ve

✗ avg(age)>45

patient

P_id	Fname	Lname	age	bg
1	Madhava	Reddy	45	o+ve
2	Abhinav	bandra	45	o-ve
3	Hari	kiran	60	b-ve
4	Mahadeva	kiran	52	o+ve
6	K_iran	kumar	39	b-ve
2	Abhinav	bandra	45	o-ve
7	Mahes%h	nambotri	36	b+ve
8	Rahul	kumar	46	b-ve
9	Bharat	kumar	56	b-ve

group by lab - 1

identify and draw the frs for the following query

```
select bg, count(*) as ‘count’ from patient
group by bg
```

group by lab-2

what is the output for the following query(must show all Intermediate groups created by se)?

```
select dept from employee group by dept
```

group by lab-3

- identify the output for the following query(must show all Intermediate groups created by se)?

```
select max(age) as 'max age',bg from patient group by bg
```

- identify the output for the following query (must show all intermediate groups created by se where ever required)?

```
select lname,MIN(age) from patient group by bg,lname
```

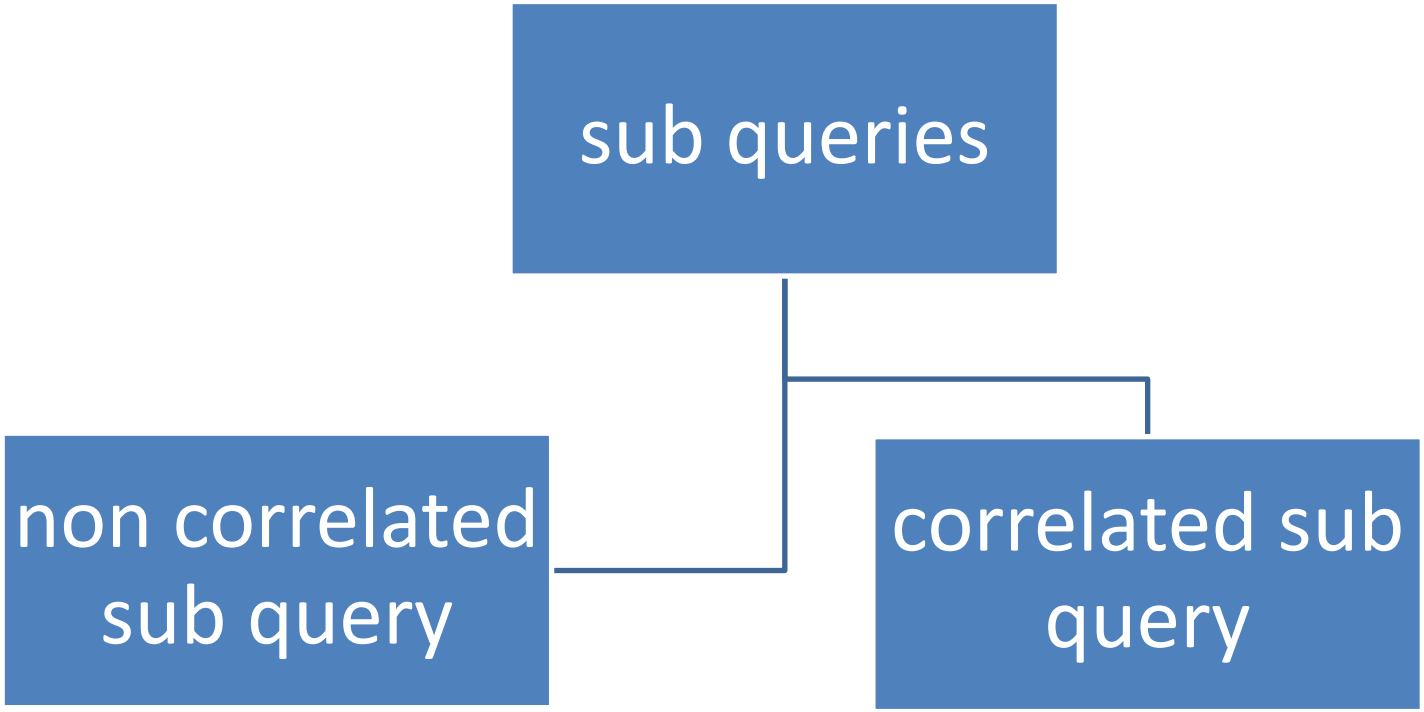
having lab-1

- identify the output for the following query?

```
select bg,MAX(age) as 'max age' from patient group by bg having MAX(age)>40
```

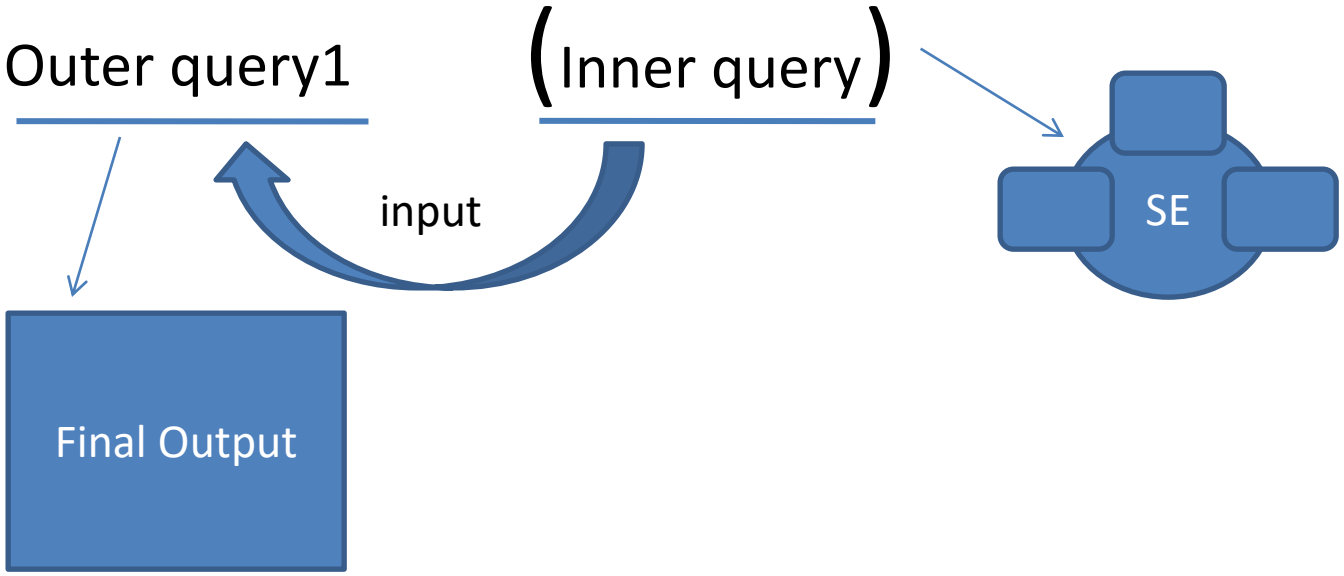
subqueries

- a subquery is a query which is usually written inside insert/update/delete/select statement
- usually inner query or sub query **must be** a select statement and outer query can be **any** t-sql statement
- sub queries usually used for identifying unknown value and the same will be substituted into outer query


sub queries
non correlated sub query
correlated sub query

non-correlated sub query

- in a non-correlated subquery , the innermost query is **executed** first.
- later inner query result will be **substituted** into outer query



non-correlated subquery sample 1

req : display fullnames of all employee's whose salary is greater than fourth **employees's** salary (whose eid is 4).

what is un known value here ?

4th employee's salary

```
select   fname+lname as 'fullname'      ?
 from employee where salary >
```

4th employee's salary

note : use sub query where value is unknown

```
select   fname+lname as 'fullname'
 from employee where salary >
        ( select salary from employee where eid=4 )
```

non-correlated subquery sample 2

req : display all employee details whose salary is greater than **'db'** department's average salary.

what is un known value here ?

db department employee's avg salary

```
select * from employee where salary >    ?
```

db dept's avg salary

note : use sub query where value is unknown

```
select * from employee where salary >
   ( select avg(salary) from employee where dept='db' )
```

non-correlated subquery sample 3

req : display all employee details whose salary is between highest paid **'.net'** dept employee's salary and least paid **db** dept employee's salary (assuming that least paid db dept salary < highest paid .net emp salary)

```
select * from employee where salary between
```

least paid db dept emp salary

and

highest paid .net dept emp salary

```
select * from employee where salary between
(select min(salary) from employee where dept='db') and
(select max(salary) from employee where dept='.net')
```

non correlated sub queries lab-1

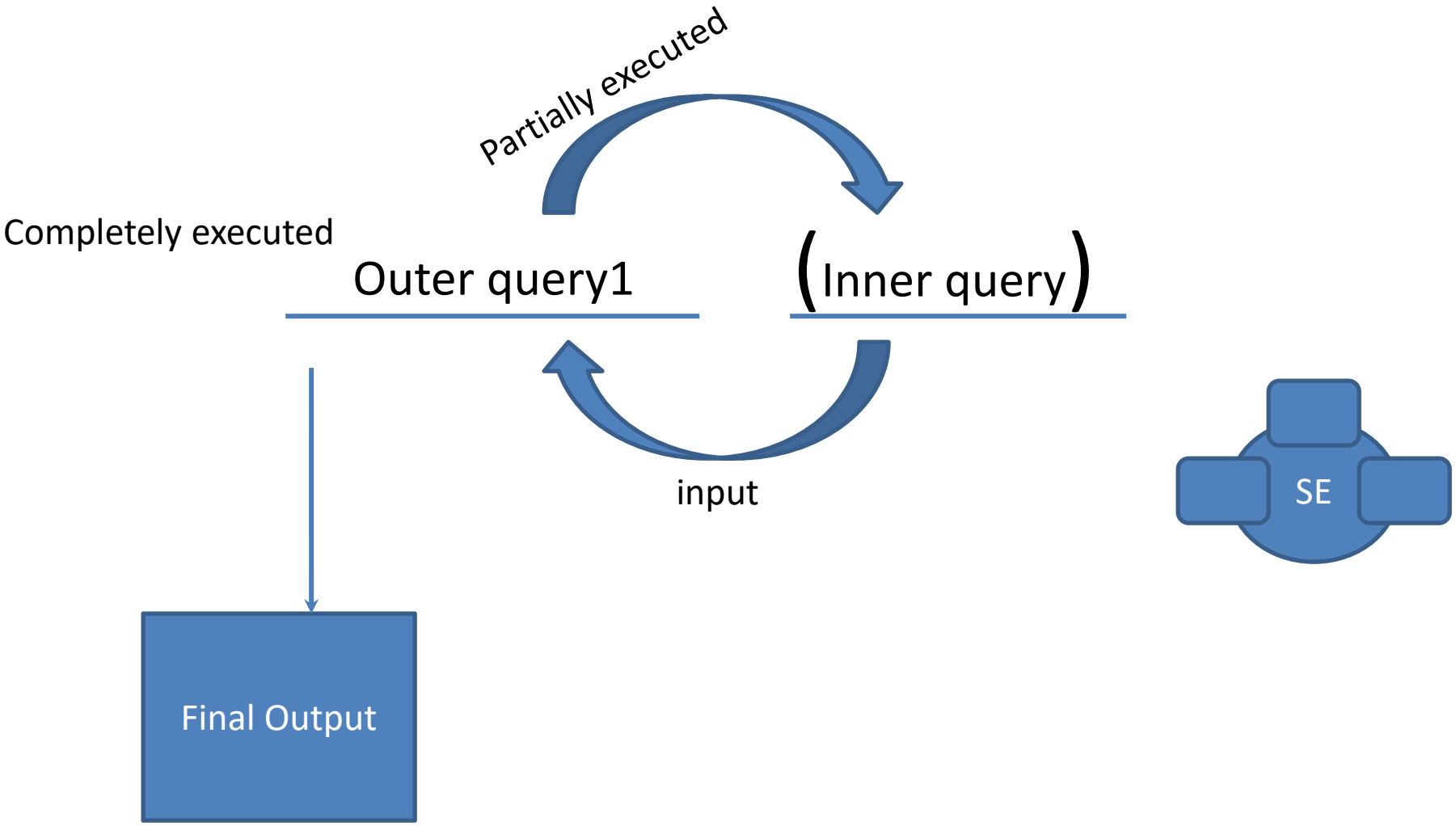
- write a query for displaying all patient details whose age is greater the age of third patient(pid=3)
- write a query for displaying all patient details whose bg is same as 6th patient's bg.
- write a query for displaying all patient details whose age is not same as 1st patients age and 3rd patients age and 9th patient's age
- find the output for the following query?

  select * from patient where age=(select age from patient where pid in (1,3,6))

correlated sub query

- correlated sub query is a type of sub query where the inner query **depends** on outer query for its **results**.
- in correlated sub query outer query **sends** some input to inner query . inner query executes and sends result to outer query. outer query will be executed and result will be added to final result set

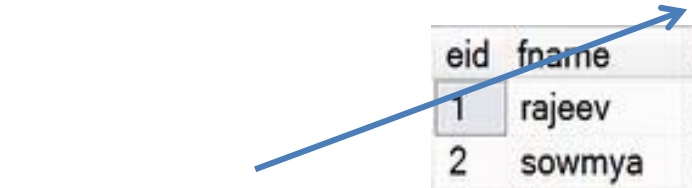
correlated sub query internals



note : to understand co-related sub queries we need to understand table aliasing first.

aliasing: is the short name given to a table.

```
select e.* from employee e where e.salary>30000
```



alias name

eid	fname	lname	age	salary	dept	doj
1	rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
3	kishore	kumar	27	36000	android	2011-10-16
4	abimanyu	biswal	22	NULL	android	2010-02-20
5	soni	kumar	24	21800	.net	2009-06-21
6	anu	_singh	22	12000	db	2010-10-23
7	_dinesh	moh%anty	23	15000	.net	2009-08-26
8	nishala	_kumari	22	18000	db	2008-07-19
1	rajeev	sukla	23	12000	.net	2011-10-23

employee e

aliasing will be useful in joins and in sub queries

aliasing is useful in co-related sub queries where we need to treat a single table as multiple tables.

correlated sub query sample

display the record which is having second highest value in c3 column

which value is the second highest ? 170

how will you find 2nd highest mathematically?

compare each value with other values.

when only **1** value is bigger

than current comparing value we can say current value as second highest

c1	c2	c3
10	AA	160
20	BB	90
3	DD	170
4	HH	280

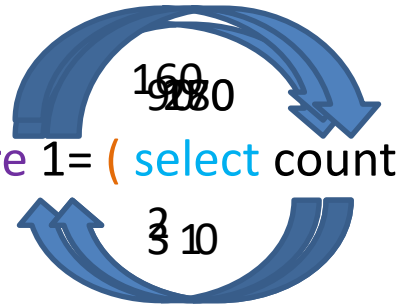
t100 t1

160 < 160	90< 160	170< 160	280< 160
160 < 90	90< 90	170< 90	280< 90
160 < 170	90< 170	170< 170	280< 170
160 < 280	90< 280	170< 280	280< 280
2	3	1	0

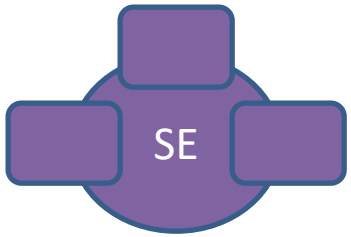
c1	c2	c3
10	AA	160
20	BB	90
3	DD	170
4	HH	280

t100 t2

correlated sub query sample



```
select t1.* from t100 t1 where 1= ( select count(*) from t100 t2 where t1.c3<t2.c3 )
```



Where 1 = 1 ☑

t1.c3 = 160	t1.c3 = 90	t1.c3 = 170	t1.c3 = 280
160 < 160	90< 160	170< 160	280< 160
160 < 90	90< 90	170< 90	280< 90
160 < 170	90< 170	170< 170	280< 170
160 < 280	90< 280	170< 280	280< 280
count (*) = 2	count(*) = 3	count(*) =1	count(*) =0

c1	c2	c3
10	AA	160
20	BB	90
3	DD	170
4	HH	280

t100 t1

c1	c2	c3
10	AA	160
20	BB	90
3	DD	170
4	HH	280

t100 t2

FRS

C1	C2	c3
3	DD	170

correlated sample 2

eid	fname	lname	age	salary	dept	doj
1	rajeev	sukla	23	12000	.net	2011-10-23
2	sowmya	kumari	23	19000	db	2010-11-13
3	kishore	kumar	27	36000	android	2011-10-16
4	abimanyu	biswal	22	NULL	android	2010-02-20
5	soni	kumar	24	21800	.net	2009-06-21
6	anu	_singh	22	12000	db	2010-10-23
7	_dinesh	moh%anty	23	15000	.net	2009-08-26
8	nishala	_kumari	22	18000	db	2008-07-19
1	rajeev	sukla	23	12000	.net	2011-10-23

employee

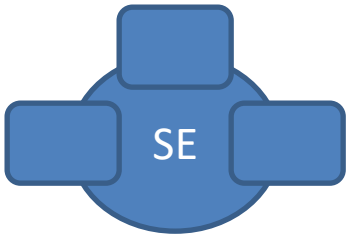
req : display second highest salary

query :

```
select e1.salary from employee e1 where 1=

(select count(*) from employee e2 where  e2.salary>e1.salary)
```

query execution

```
select e1.salary from employee e1 where 1 = 0
                                         1
    (select count(*) from employee e2 where  e2.salary>e1.salary)
```



e2.salary

>

e1.salary

salary
12000
19000
36000
NULL
21800
12000
18000
15000
12000

salary
12000
19000
36000
NULL
21800
12000
18000
15000
12000

FRS

salary
21800

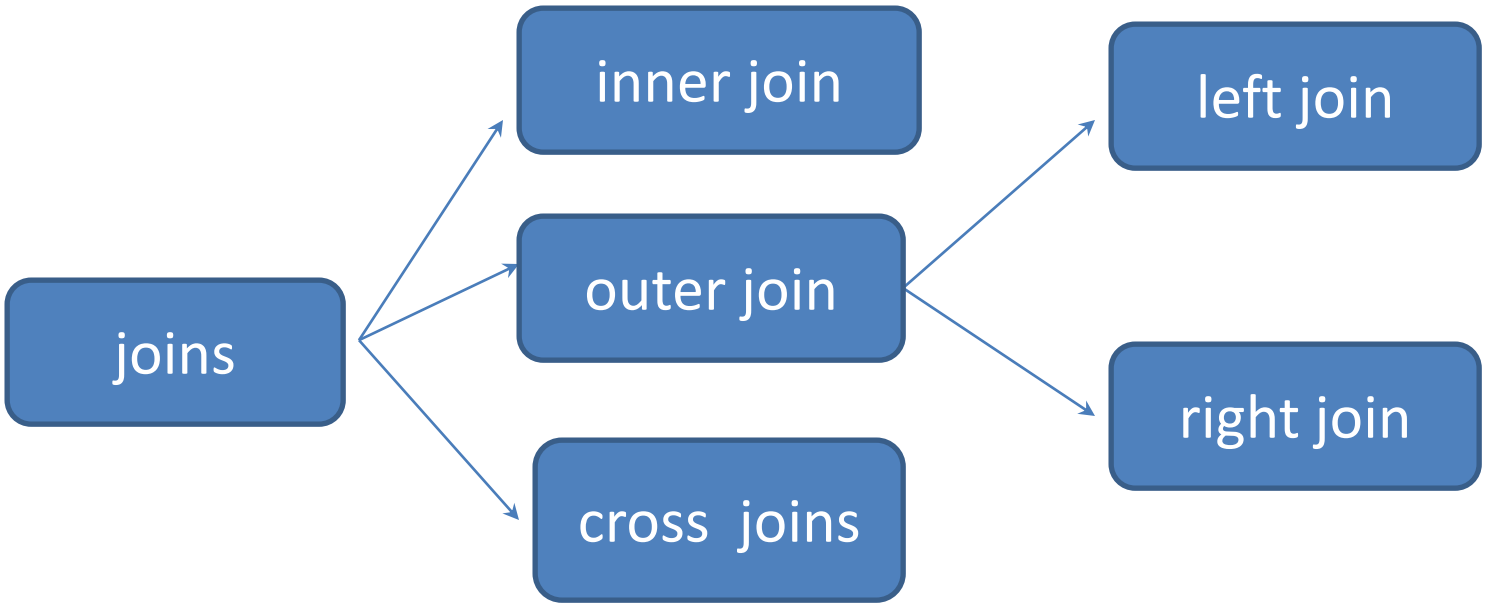
correlated sub query lab

- identify the output for the following query (and also write detailed analysis for the same)?

```
select p1.* from patient p1 where 3=( select count(distinct(p2.pid)) from patient p2 where p2.age>p1.age)
```

joins

using joins we can fetch data from one or more tables into a result set.



inner join

in inner joins only the matched records(based on condition) from left side table and right side table are added to result set

syntax:

.....left_table_name alias_name inner join/join right_table_name alias_name **on** join_condition

inner join/ join sample

by looking at these tables ,
tell me what is madhav's bloodgroup

how did you identified

by comparing

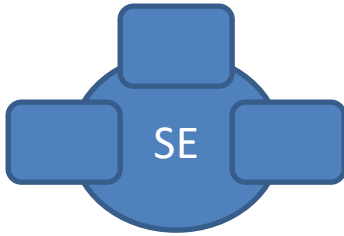
patient1 p

pid	name	bg_id	age
1	madhav	1	24
2	hari	2	27
3	kiran	1	21
4	raj	3	29

bg b

id	bg
1	o+ve
2	o-ve
3	b+ve
4	b-ve
5	ab-ve

```
select p.name, b.bg, p.age from patient1 p  join bg b  on p.bg_id=b.id
```



final result set

name	bg	age
madhav	o+ve	24
hari	o-ve	27
kiran	o+ve	21
raj	b+ve	29

ERS

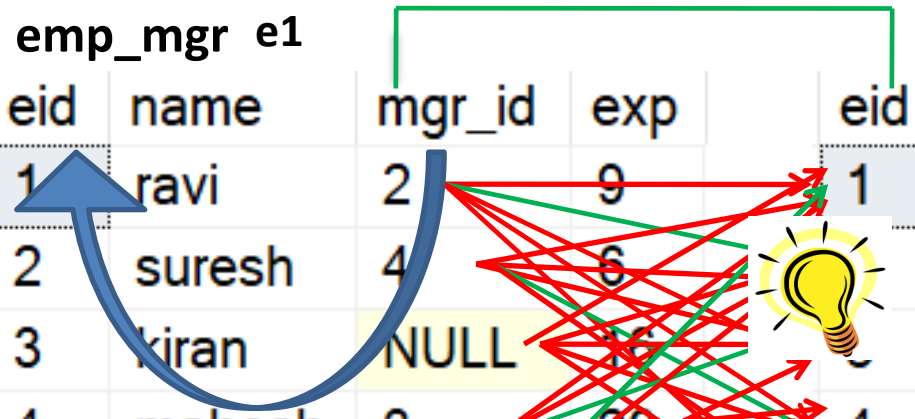
name	bg	age
madhav	o+ve	24
kiran	o+ve	21
hari	o-ve	27
raj	b+ve	29

emp_mgr e1

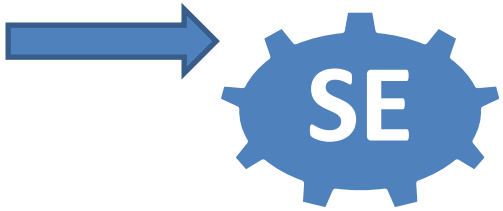
eid	name	mgr_id	exp
1	ravi	2	9
2	suresh	4	6
3	kiran	NULL	16
4	mahesh	3	26
5	hari	1	3

emp_mgr e2

eid	name	mgr_id	exp
1	ravi	2	9
2	suresh	4	6
3	kiran	NULL	16
4	mahesh	3	26
5	hari	1	3



```
select e1.name as 'employee', e2.name as 'manager' from emp_mgr e1
join emp_mgr e2 on e1.mgr_id=e2.eid
```



employee	manager
ravi	suresh
suresh	mahesh
mahesh	kiran
hari	ravi

for comparing a table to itself you need to **consider that table as 2 separate tables using 2 different alias names**

take a look on table and tell me , who is manager for employee **ravi**

by relating the mgr_id and eid columns we can tell the managers of the employees

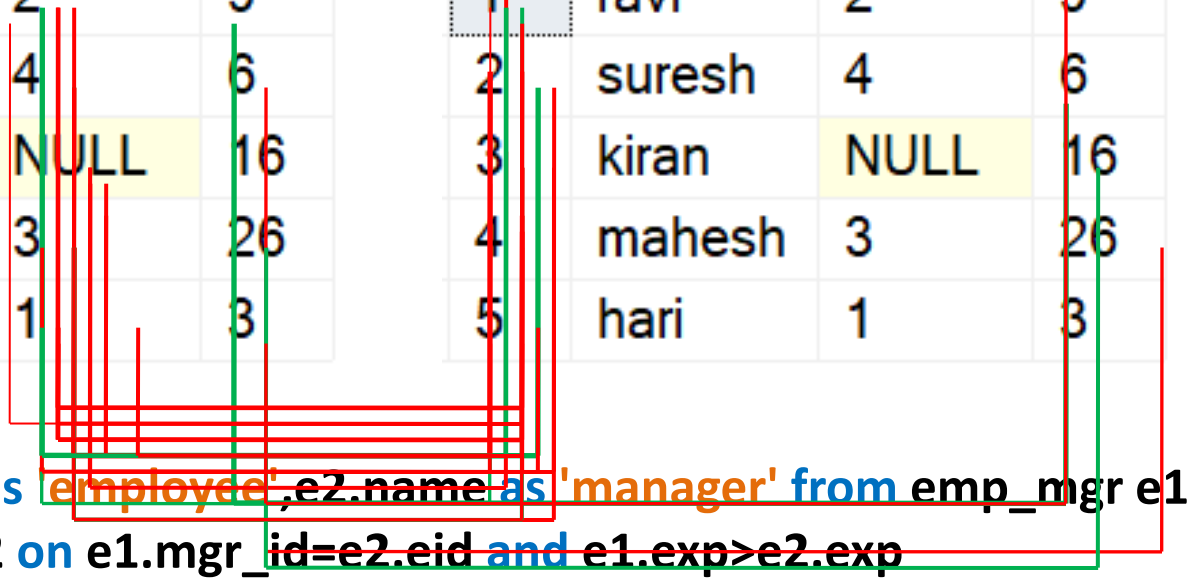
req: display all employee and manager names , where the employee's experience is greater than their manager's

emp_mgr e1

eid	name	mgr_id	exp
1	ravi	2	9
2	suresh	4	6
3	kiran	NULL	16
4	mahesh	3	26
5	hari	1	3

emp_mgr e2

eid	name	mgr_id	exp
1	ravi	2	9
2	suresh	4	6
3	kiran	NULL	16
4	mahesh	3	26
5	hari	1	3



```
select e1.name as 'employee',e2.name as 'manager' from emp_mgr e1
join emp_mgr e2 on e1.mgr_id=e2.eid and e1.exp>e2.exp
```

employee	manager
ravi	suresh
mahesh	kiran

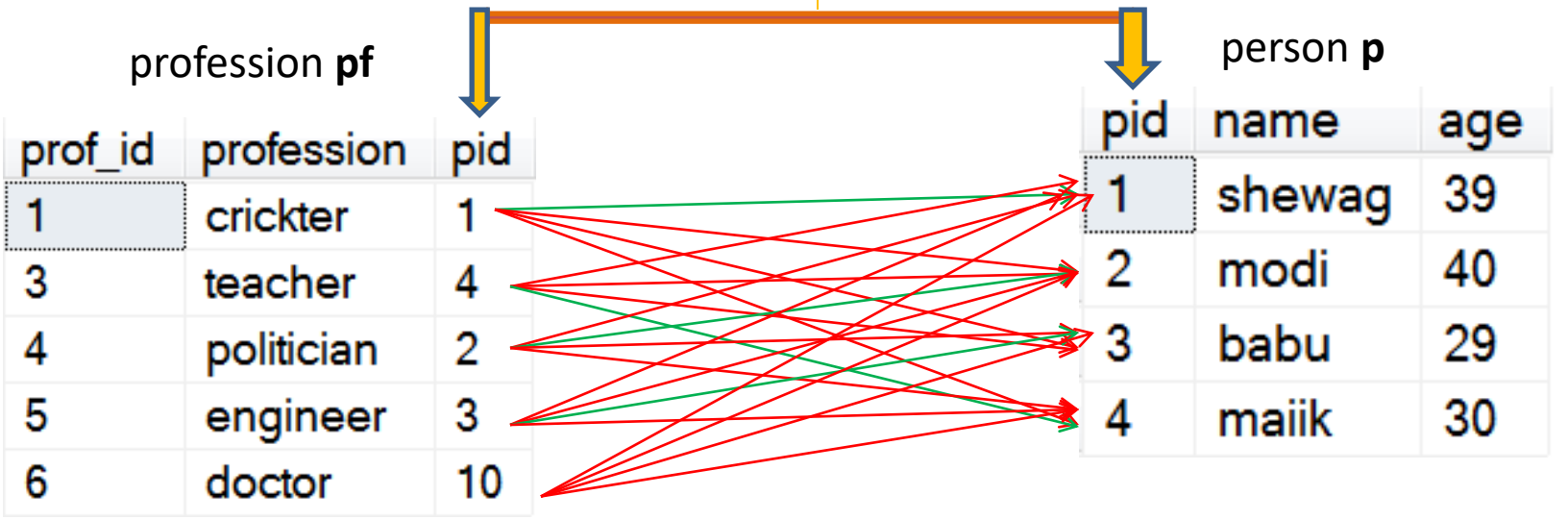
left outer join

in a left outer join , **all** the **data** from left table will be **added** to the **result set** but **only** the matched **records** from the **right** table is included to the **result set** .

wherever there is **no match** in the right table , **null** values are **added** to the **result set**

syntax:

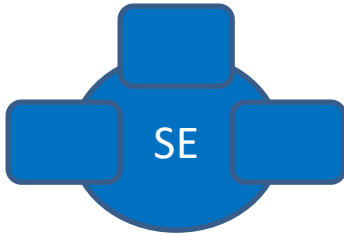
…… left_table_name alias _name **left outer join**
right_table_name alias_name **on** condition




```
select  pf.* , p.* from  profession pf  left outer join person p  on  pf.pid = p.pid
```

final result set

prof_id	profession	pid	pid	name	age
1	crickter	1	1	shewag	39
3	teacher	4	4	maiik	30
4	politician	2	2	modi	40
5	engineer	3	3	babu	29
6	doctor	10	NULL	NULL	NULL



in left outer join, all left table data must be added to result set

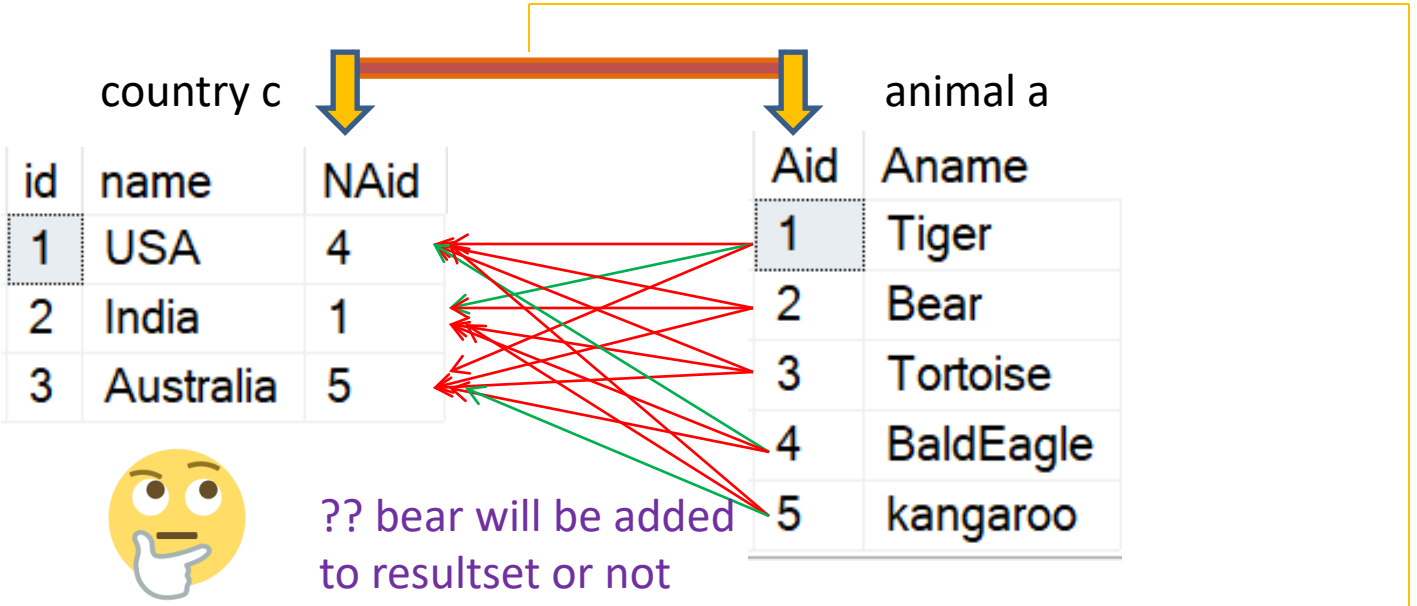
right outer join

in a **right** outer join , **all** the **data** from right table will be **added** to the **result set** , and **only** the matched records from the **left table** is added to result set .

wherever there is **no match** in the **left table** , **null** values are **added** to the result set

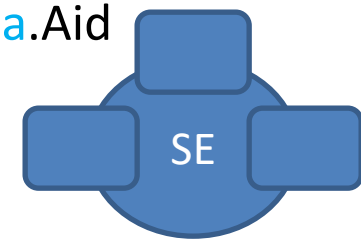
syntax :

…. left_table_name alias _name **right outer join**
right_table_name alias_name **on** condition



req: now I would like to display **all animal names** along with country names if any **matching countries** are present

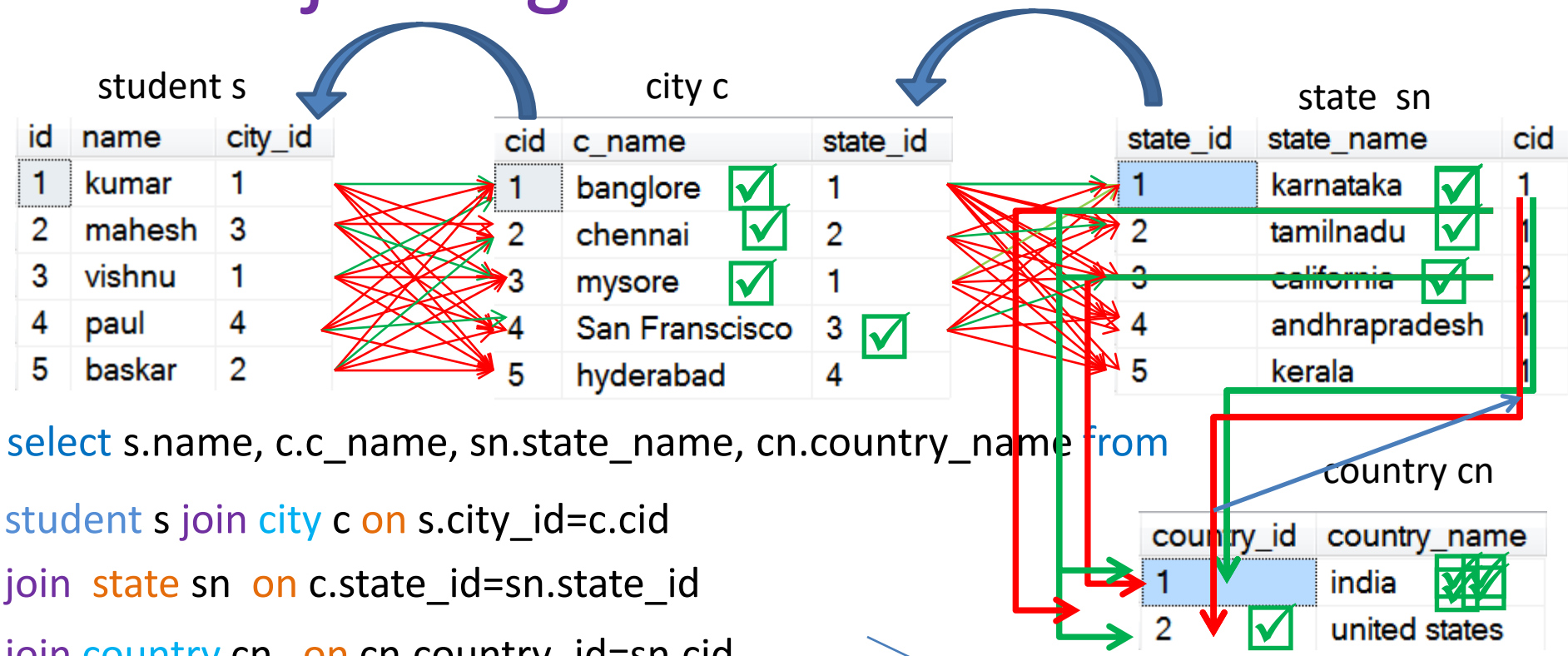
```
select c.name, a.aname from country c right join animal a on c.NAid=a.Aid
```



final result set

name	aname
India	Tiger
NULL	Bear
NULL	Tortoise
USA	BaldEagle
Australia	kangaroo

joining 3 or more tables



id	name	city_id
1	kumar	1
2	mahesh	3
3	vishnu	1
4	paul	4
5	baskar	2

cid	c_name	state_id
1	banglore	1
2	chennai	2
3	mysore	1
4	San Franscisco	3
5	hyderabad	4

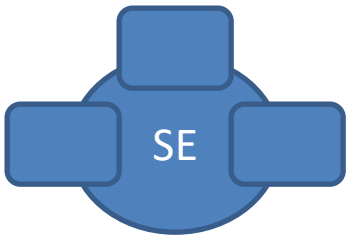
state_id	state_name	cid
1	karnataka	1
2	tamilnadu	1
3	california	2
4	andhrapradesh	1
5	kerala	1

country_id	country_name
1	india
2	united states

```
select s.name, c.c_name, sn.state_name, cn.country_name from
student s join city c on s.city_id=c.cid
join  state sn  on c.state_id=sn.state_id
join country cn   on cn.country_id=sn.cid
```

FRS ERS

name	c_name	state_name	country_name
kumar	banglore	karnataka	india
mahesh	mysore	karnataka	india
vishnu	banglore	karnataka	india
paul	San Franscisco	california	united states
baskar	chennai	tamilnadu	india



cross join

cross join with join condition will produce same result as inner join

cross join without join condition will produce the cartesian products of the tables which are involved in join

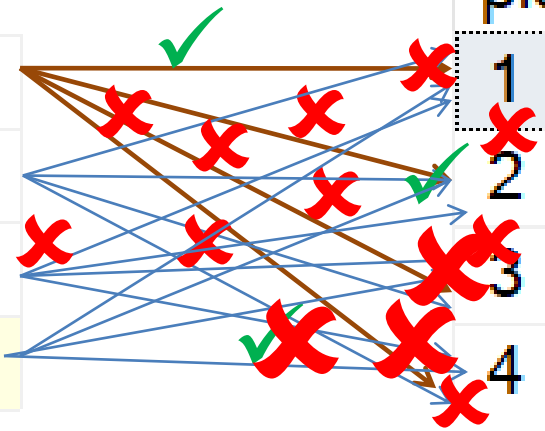
we must use cross join keyword for cross join

profession pf

SiNo	profession	pid
1	Cricketer	1
3	teacher	4
4	Politician	2
5	scientist	NULL

person p

pid	name	age
1	sehwag	39
2	modi	40
3	babu	29
4	malik	30



select pf.*,p.* from person p cross join profession pf on pf.pid=p.pid

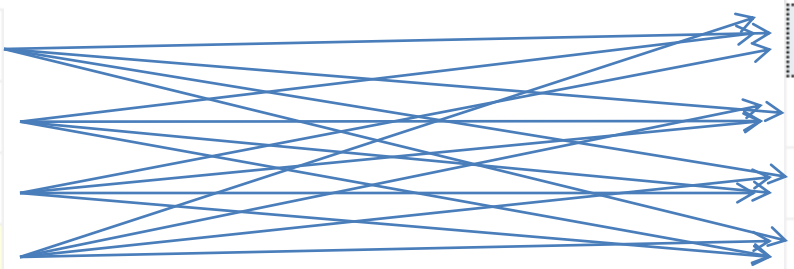
FRS

SiNo	profession	pid	pid	name	age
1	Cricketer	1	1	sehwag	39
3	teacher	4	4	malik	30
4	Politician	2	2	modi	40

cross join without join condition

profession pf

SiNo	profession	pid
1	Cricketer	1
3	teacher	4
4	Politician	2
5	scientist	NULL



person p

pid	name	age
1	sehwag	39
2	modi	40
3	babu	29
4	malik	30

select pf.*, p.* from person p cross join profession pf

pid	name	age	sino	profession	pid
1	crickter	1	1	shewag	39
1	crickter	1	2	modi	40
1	crickter	1	3	babu	29
1	crickter	1	4	maiik	30
3	teacher	4	1	shewag	39
3	teacher	4	2	modi	40
3	teacher	4	3	babu	29
3	teacher	4	4	maiik	30
4	politician	2	1	shewag	39
4	politician	2	2	modi	40
4	politician	2	3	babu	29
4	politician	2	4	maiik	30
5	scientist	NULL	1	shewag	39
5	scientist	NULL	2	modi	40
5	scientist	NULL	3	babu	29
5	scientist	NULL	4	maiik	30

joins samples

c1	c2	c3
10	AA	6
20	BB	80
30	CC	260
4	DD	-60

T1

c4	c5
1	RR
20	JJ
60	KL
4	NM

T2

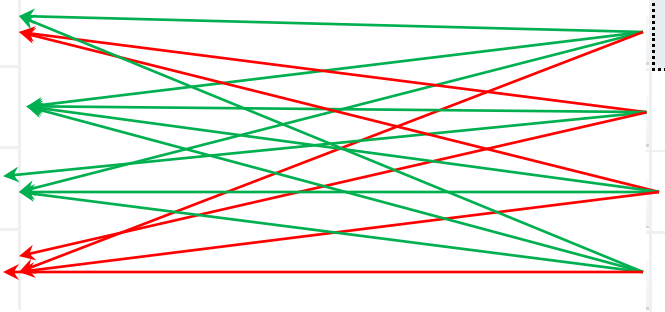
```
select t1.* ,t2.* from T1 t1 inner join T2 t2 on t1.c3>t2.c4

select t1.* ,t2.* from T1 t1 inner join T2 t2 on t1.c1!=t2.c4

select t1.* ,t2.* from T1 t1 left outer join T2 t2 on t1.c1!=t2.c4

select t1.* ,t2.* from T1 t1 right outer join T2 t2 on t1.c1!=t2.c4

select t11.* , t1.* from T1 t11 join T1 t1 on 3*t11.c3<t1.c1
```

T1 t1

c1	c2	c3
10	AA	6
20	BB	80
30	CC	260
4	DD	-60

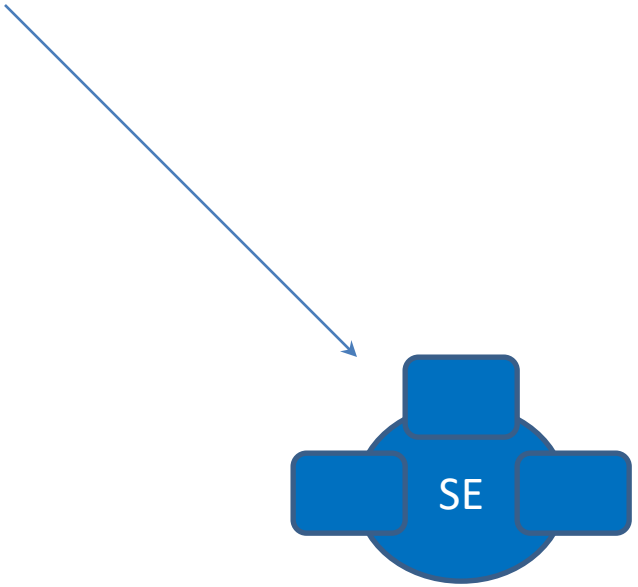
T2 t2

c4	c5
1	RR
20	JJ
60	KL
4	NM

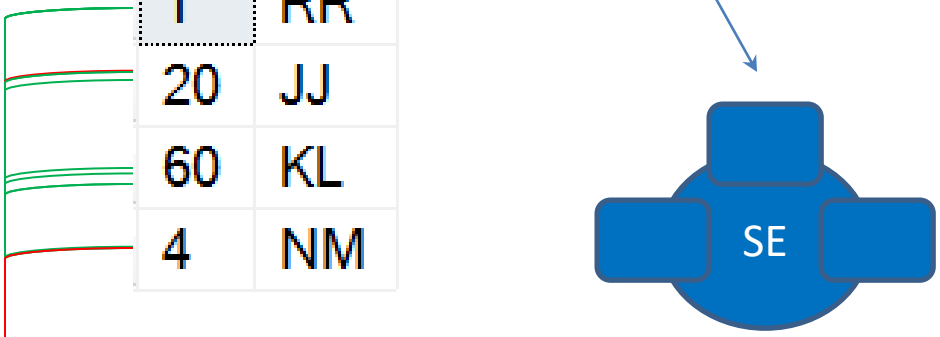


```
select t1.* ,t2.* from T1 t1 inner join T2 t2 on t1.c3>t2.c4
```

c1	c2	c3	c4	c5
10	AA	6	1	RR
20	BB	80	1	RR
30	CC	260	1	RR
20	BB	80	20	JJ
30	CC	260	20	JJ
20	BB	80	60	KL
30	CC	260	60	KL
10	AA	6	4	NM
20	BB	80	4	NM
30	CC	260	4	NM



```
select t1.* ,t2.* from T1 t1 inner join T2 t2 on t1.c1!=t2.c4
```



T2 t2

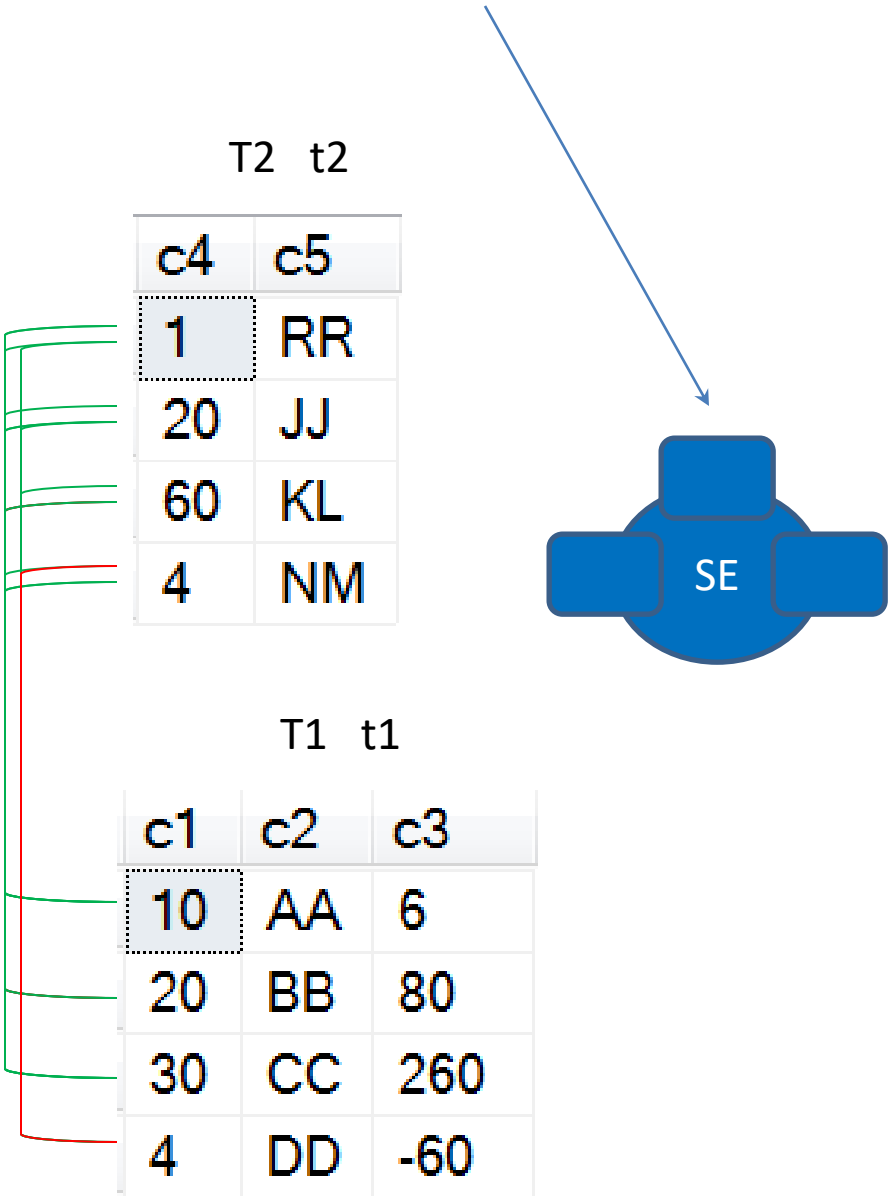
c4	c5
1	RR
20	JJ
60	KL
4	NM

T1 t1

c1	c2	c3
10	AA	6
20	BB	80
30	CC	260
4	DD	-60

c1	c2	c3	c4	c5
10	AA	6	1	RR
20	BB	80	1	RR
30	CC	260	1	RR
4	DD	-60	1	RR
10	AA	6	20	JJ
30	CC	260	20	JJ
4	DD	-60	20	JJ
10	AA	6	60	KL
20	BB	80	60	KL
30	CC	260	60	KL
4	DD	-60	60	KL
10	AA	6	4	NM
20	BB	80	4	NM
30	CC	260	4	NM

```
select t1.* ,t2.* from T1 t1 left outer join T2 t2 on t1.c1!=t2.c4
```



c1	c2	c3	c4	c5
10	AA	6	1	RR
10	AA	6	20	JJ
10	AA	6	60	KL
10	AA	6	4	NM
20	BB	80	1	RR
20	BB	80	60	KL
20	BB	80	4	NM
30	CC	260	1	RR
30	CC	260	20	JJ
30	CC	260	60	KL
30	CC	260	4	NM
4	DD	-60	1	RR
4	DD	-60	20	JJ
4	DD	-60	60	KL

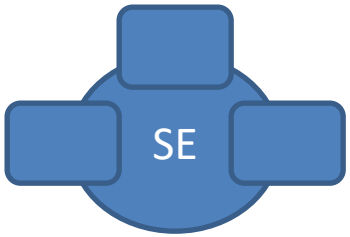
```
select t1.* ,t2.* from T1 t1 right outer join T2 t2 on t1.c1!=t2.c4
```

T2 t2

c4	c5
1	RR
20	JJ
60	KL
4	NM

T1 t1

c1	c2	c3
10	AA	6
20	BB	80
30	CC	260
4	DD	-60



c1	c2	c3	c4	c5
10	AA	6	1	RR
20	BB	80	1	RR
30	CC	260	1	RR
4	DD	-60	1	RR
10	AA	6	20	JJ
30	CC	260	20	JJ
4	DD	-60	20	JJ
10	AA	6	60	KL
20	BB	80	60	KL
30	CC	260	60	KL
4	DD	-60	60	KL
10	AA	6	4	NM
20	BB	80	4	NM
30	CC	260	4	NM

```
select t11.* , t1.* from T1 t11 join T1 t1 on 3*t11.c3<t1.c1
```

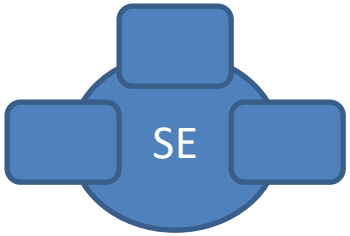
c1	c2	c3
10	AA	6
20	BB	80
30	CC	260
4	DD	-60

T1 t11

3*c3
18
240
780
-180

c1	c2	c3
10	AA	6
20	BB	80
30	CC	260
4	DD	-60

T1 t1



c1	c2	c3	c1	c2	c3
10	AA	6	20	BB	80
10	AA	6	30	CC	260
4	DD	-60	10	AA	6
4	DD	-60	20	BB	80
4	DD	-60	30	CC	260
4	DD	-60	4	DD	-60

tables required for joins lab

fp_id	f_name	l_name	state_id
1	apj	abdul kalam	2
2	nr	narayana murthy	1
3	ratan	tata	5

famous_people

s_id	s_name
1	Karnataka
2	Tamilnadu
3	Uttar Pradesh
4	Madhya Pradesh
5	Maharashtra

state

joins lab 1

- identify the output for the following queries?

```
select  fp.f_name, fp.l_name, s.s_name from famous_people fp
inner join state s on fp.state_id=s.s_id
```

```
select  fp.f_name, fp.l_name, s.s_name from famous_people fp
inner join state s on fp.state_id!=s.s_id
```

```
select concat(fp.f_name,' ',fp.l_name,' is not from ', s.s_name) as result
from famous_people fp inner join state s on fp.state_id != s.s_id
```

joins lab 2

- identify the output for the following queries?

```
select  fp.f_name, fp.l_name, s.s_name from famous_people fp
left join state s on fp.state_id=s.s_id
```

```
select  fp.f_name, fp.l_name, s.s_name from famous_people fp
right join state s on fp.state_id=s.s_id
```

```
select  fp.f_name, fp.l_name, s.s_name from famous_people fp
right join state s on fp.state_id>s.s_id
```

joins lab 3

eid	name	mgr_id	exp
1	ravi	2	9
2	suresh	4	6
3	kiran	NULL	16
4	mahesh	3	26
5	hari	1	3

employee_manager

- write a query for finding all employee names whose exp is greater than their managers exp?
- write a query for displaying all employee names along with their manager names if any (the result set must also contain kiran)

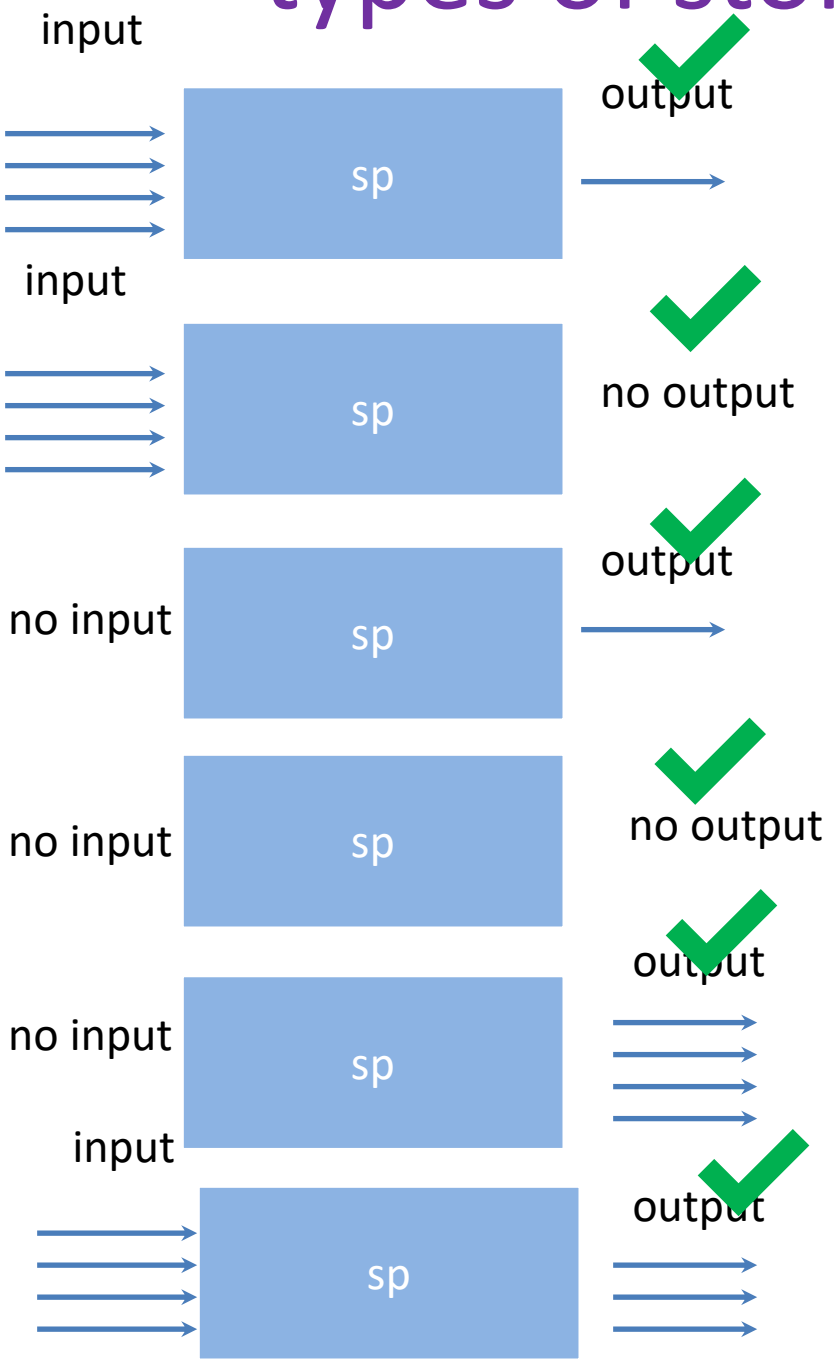
stored procedures

- stored procedure is almost same as function/method in normal **programming** languages.
- a stored procedure is a compiled query **(query whose execution plan is cached)**
- a sp can take 0 or more inputs and can return 0 or more outputs.

stored procedures advantages

- stored procedures execution is **faster** compared to in-line queries,since stored procedures are treated as compiled queries
- **compiled query : a query whose execution plan is cached**
- stored procedures **reduces** the network traffic
- stored procedures **improves** the **security** by hiding the underlying tables

types of stored procedures



output:

1) **using select statements**
2) **output parameters**
3) **return statement**

note: we can return only integer values using return statement

sp syntax

syntax :

delimiter custom delimiter
create procedure procedurename(optional parameters)
begin
// **procedure** body...;
// **procedure** body...;
end;
custom delimiter

note : custom delimiter can be // or $$

calling sp:
call procedurename **(inputval1,inputval2,.....)**

stored procedure sample 1

write a sp with the name getpatient to get all patient details from table patient.

```
delimiter $$
create procedure getpatient()
begin
select * from patient;
end;
$$
```

calling sp:

```
call getpatient()
```

output

	pid	fname	lname	age	bg
1	1	Madhava	Reddy	45	O+ve
2	2	Abhinav	ban...	45	O-ve
3	4	Hari	Kiran	60	B-ve
4	3	Madhava	Kiran	52	O+ve
5	5	veena	kum...	42	NULL
6	6	K_iran	Kum...	39	B-ve
7	2	Abhinav	ban...	45	O-ve
8	7	Mahes...	Nam...	36	B+ve
9	8	Rahul	Kum...	46	B-ve
10	9	bharat	Kum...	56	B-ve

stored procedure sample 2

write a sp with the name insertemp for inserting new employee details into employee table

```
delimiter $$
create procedure insertemp
(
  eid int,
  fname varchar(20),
  lname varchar(20),
  age int,
  sal int,
  dept varchar(20),
  doj date
)
begin
  insert into employee values(eid,fname,lname,age,sal,dept,doj);
end;
$$
```

calling sp:

```
call insertemp(10,'harish','rao',23,25000,'android','2022-01-10')
```

stored procedure lab-1

- write a sp with the name insertpatient for inserting new patient record into patient table

stored procedure lab2

write a stored procedure for inserting new record into the patient table as per the following condition.

1. sp must take 4 input parameters fname, lname, age , bg and the pid must be current max pid + 1

   (use sub query for findng cur max pid)

stored procedure lab3

1. write a stored procedure to delete a record from patient table for a given pid value.

   take 1 input parameter to get pid value.

2. write a stored procedure to update a record from patient table for a given pid value based on following conditions (sp must take 5 parameters)

   pid , new_fn, new_ln, new_age, new_bg

loadable function

- loadable function is similar to a stored procedure
- loadable function contains set of compiled my-sql statements (which is similar to stored procedures)
- loadable fun
  parameters
- loadable fu
  which alter
- loadable fu
  procedure

students must remember this theory (not used in most of the application development) but can be asked in interviews only theory

loadable function vs stored procedure

stored procedure	loadable function
supports input & output parameters	supports only input parameters
can write any type of sql queries	can't write sql queries which modifies state of db (ex. insert/update/delete/create etc..)
stored procedure can call loadable function	loadable function can't call stored procedure

views

- a view is a virtual table or a stored query
- a view must **contain** only one select statement
- a view can internally refer 1 or more table(s) or views
- a view provides security to tables

types of views :-

1. updatable view
2. non-updatable view

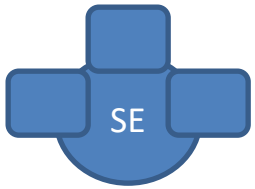
syntax :

```
create view viewname
as
select   statement
```

view sample

now I would like to create a view for this table

```
create view vpatient
as
select * from patient
```




```
select * from vpatient

insert into vpatient values (5,'suresh','babu','o+ve')

select * from vpatient

update vpatient set fname='rakesh' where pid=5
```

patient

pid	Fname	Lname	bg
1	ravi	kiran	o+ve
2	bhinav	bandra	o-ve
3	abishek	kumar	b+ve
4	mahesh	suri	o+ve
5	rakesh	babu	o+ve

vpatient

pid	Fname	Lname	bg
1	ravi	kiran	o+ve
2	bhinav	bandra	o-ve
3	abishek	kumar	b+ve
4	mahesh	suri	o+ve
5	rakesh	babu	o+ve

This is an example for Updatable view

non-updatable view

we **can't** update **data** in underlying table by using a **non-updatable** views

a view will become non-updatable view in following conditions

1. when **select list** of a view **contains distinct** keyword
2. when **select list contains aggregate function**
3. when **select list** includes **group by** clause

we will understand non-updatable views, with a sample table

patient

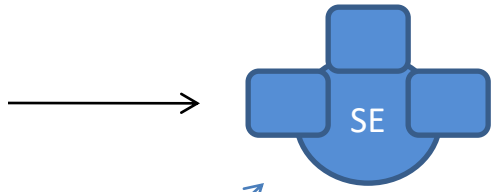
pid	Fname	Lname	bg
1	ravi	kiran	o+ve
2	bhinav	bandra	o-ve
3	abishek	kumar	b+ve
4	mahesh	suri	o+ve

nuvpatient

pid	Fname	Lname	bg
1	ravi	kiran	o+ve
2	bhinav	bandra	o-ve
3	abishek	kumar	b+ve
4	mahesh	suri	o+ve
5	rakesh	babu	o+ve

```
create view nuvpatient
as
select distinct * from patient
```

```
select * from  nuvpatient
```




```
update  nuvpatient  set fname='raju' where pid=1
```

ERROR

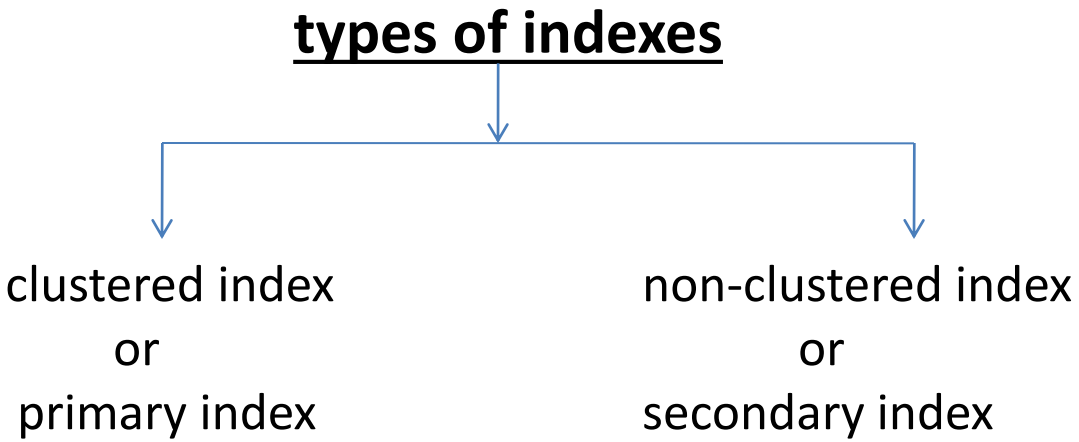
```
create view pt
as
select bg from patient group by bg
```

```
update  nuvpatient  set fname='raju' where pid=1
```

ERROR

indexes

- using indexes we can quickly find the information from a table or from an indexed view



clustered index

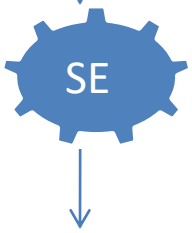
- in a clustered index , the actual table is stored in the leaf pages of b-tree *[binary tree]*
- only 1 clustered index is possible per table

note:
if your table is having primary key constraint, clustered index is created automatically

when to use index

for understanding why to use indexes , we will consider this patient table, assuming that 1,00,000 records are present in patients table .

select * from patients where p_id=99999



SE has to perform approximately 99999 search operations to retrieve data

where p_id=99999

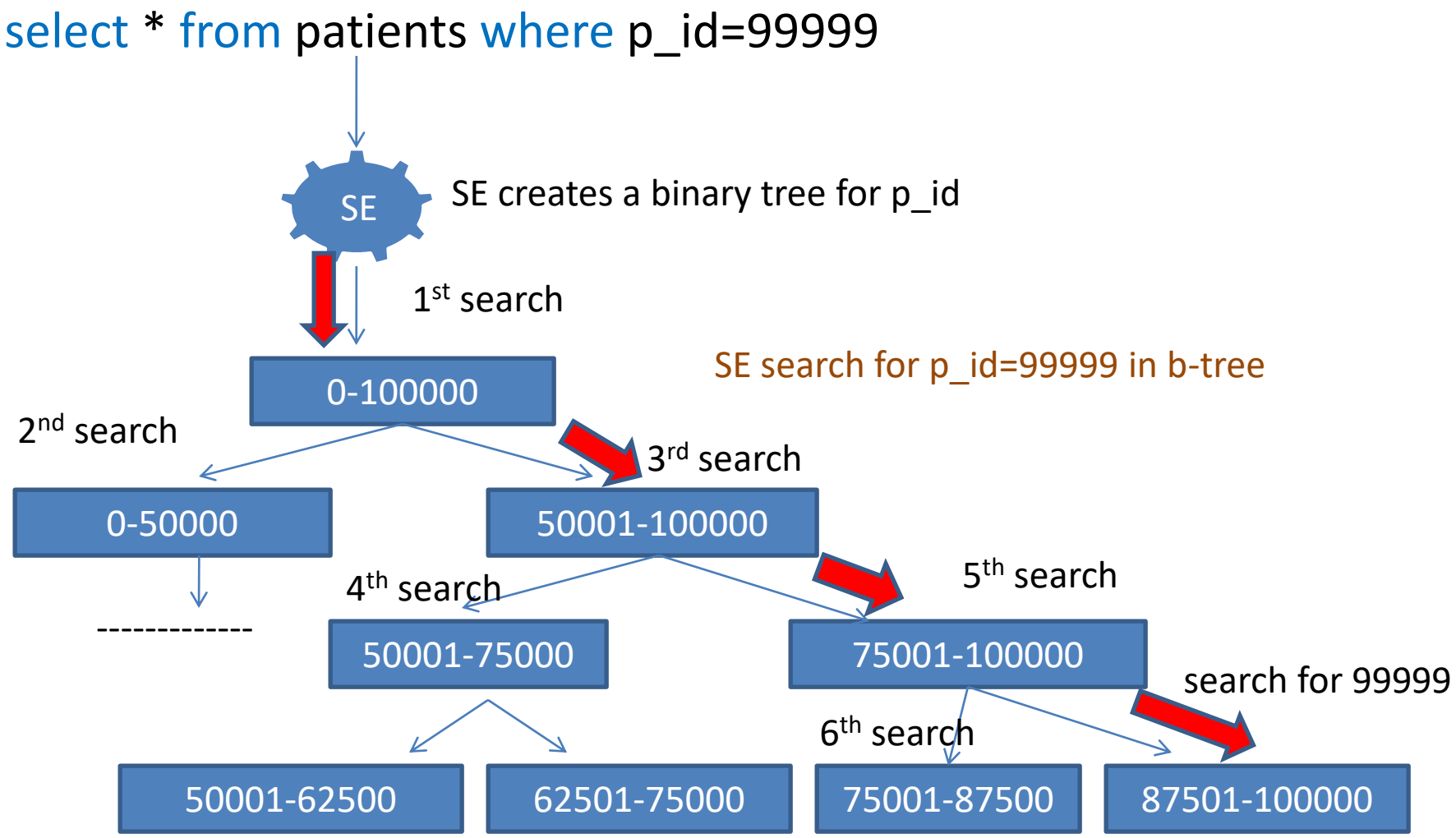
p_id	Fname	Lname	Age	Bg
1	Madhava	Reddy	45	O+ve
3	Madhava	Kiran	52	O+ve
4	Hari	Kiran	60	B-ve
5	veena	kumari	42	NULL
6	K_iran	Kumar	39	B-ve
7	Mahes%h	Nambotri	36	B+ve
8	Rahul	Kumar	46	B-ve
9	BHARAT	Kumar	56	B-ve
----	--------	---------	--	--------
7700	ravi	kumar	49	o-ve
99800	rahul	kumar	56	b-ve
99999	zubi	kumari	24	b+ve
100000	rubina	begum	48	ab+ve

output

p_id	Fname	Lname	Age	Bg
99999	zubi	kumari	24	b+ve

this operation is called as table scanning ,
and this is not good approach for searching records

searching on a clustered index



total number of searches are 7+12498(99999-87501) =12505

total searches are reduced compared to 99999 searches

non clustered index

- non clustered index contains **pointers** to the actual table
- we can create **more than one** non clustered index per table

note:

if your table is having unique constraint, non clustered index is created automatically

imp interview questions

- what is a transaction ?
  - transaction is used for grouping logically **related** queries as a single unit
  - transaction must always follow acid properties (atomicity consitancy isolation durability)
- what is a trigger ?
  - trigger is like a function which will be automatically executed for users actions like insert , update , delete