

SQL SERVER

Class notes

By

Mr . Kanna Babu sir



**SRI RAGHAVENDRA
XEROX**

Software languages Material Available

Beside Bangalore Iyyager Bakery .Opp. CDAC, Balkampet Road,
Ameerpet,Hyd

9951596199

10-11-2014

SQL - SERVER

* Front End application: The application where exactly the user will interact is called frontend application. it can be developed by frontend tools like ex:- PHP, .net, Java.
ex:- gmail, inbox, drafts etc.

* Back end Application: The application where exactly user data is maintained and stored i.e
→ it can be developed by backend tools like MSSQL SERVER, ORACLE, MySQL, EXCEL, ACCESS, DB2 etc
.net, MSBI, testing tools — SQL SERVER.
Java - oracle., PHP - MySQL

~~a sy.~~ 11-11-2014 Database is systematic collection of data.
since the data in the database is organised. it makes the data management easy.
ex:- satya, mobile shoppe, payroll, university.

* DBMS: is a data base management system.
it is a software which is used to manage data base.
ex:- EXCEL, ACCESS, FOX PRO.

* RDBMS: It is a relational database management system which is used to manage DB.

ex:- SQL SERVER, ACCESS & ORACLE, MySQL

* SQL: is a structured query language

SQL → SEQUEL → structured English query language.

SQL is a language which is used to communicate with database, i.e. if we want to store the data or manipulate the data we have to use SQL.

* In database the data will be stored in the form of table.

@ what is table?

Table is a database object which is used to store the data in the form of rows and columns.

eno	ename	sal
101	anil	20,000
102	sunil	30,000
103	teja	40,000

row (or) record

cell or tuple

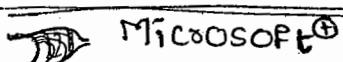
* whenever we are storing the data we should two (a)

- ① single cell must consists of single value
- ② Each Record must be unique.

Steps to open SQL SERVER management studio.

start
↓

MSSQLSERVER-2008 → SQL SERVER management studio.



SQL SERVER 2012

server type : Database engine

server name : computer name / IP address.

authentication : SQL Server Authentication

login : sa

password :

server type: ① Data base engine

② Reporting services

③ Integration services

④ Analysis services

→ Business intelligence the i.e. the high level

authorities in companies like CEO, Project Manager, etc for decision making we use SSIS, SSAS, SSRS.

→ SSIS → is used to collect data from datasource.
ex:- XML, Database.

→ SSAS is used to analyse the data

→ SSRS is used to view the data, reporting tool.

collect

SSIS

Analyse

SSAS

Reporting

SSRS

As a .NET developer we are using SQL data base as data

(a) Server Name: Here we have to mention the name of the computer where SQL Server software was installed.

Server name = computer name / IP address / - / local host.

Authentication: It is a process of checking user credentials.

User credentials means username and password. Any user who is having username and password is called as authenticated user.

We can connect to SQL Server database in two ways

① Windows Authentication

② SQL Server Authentication.

When ever we are connecting to SQL Server database using Windows Authentication it is not required to give username and password, because by default you are the valid user of Windows O.S.

→ When ever we connect to SQL Server database using SQL Server Authentication we have to mention username and password.

→ The default username is sa and password is abc.

* Database: - is systematic collection of data.

① Predefined database

② User defined database.

① Predefined database: - The database that we are installed when we install SQL MS those databases are called pre-defined (i)

~~user defined~~
systems

② Master: - The configuration settings that are required to run SQL Server Management Studio are available in master database.

it maintains server name, authentication details like username, password etc.

→ It is always recommended to maintain a separate copy of master database in SQL Server MS.

⑥ Model Database: is a template for all the other databases.

→ whenever we are creating user-defined database then the contents of the model database ~~will~~ will be copied into user-defined database.

⑦ Tempdb: it is used to store temporary tables and stored procedures.

⑧ MSDB: it is used to schedule jobs and alerts.

13-11-14

* User-defined database: The database that are created depending on the user requirement are called as user-defined databases.

* Language Commands: These are used to create database or data

Ex:-

DDL DML DRL TCL OCL

① Data definition → Drop, create, alter, truncate
manipulate

② Data management → update, delete, insert

③ Data retrieve language → select,

④ TCL - Transaction control → commit, rollback, savepoint

⑤ DCL → Revoke, Grant.

Data control language.

Language Commands

DDL

↓

Drop

Create

Alter

Truncate

DML

↓

update

delete

insert

TCL

↓

commit

rollback

save point

DRL

↓

select

OCL

↓

Grant

Revoke

deny

②

DDL:- Data definition language commands.

These commands are used to create

- ① Create database, database objects like, tables, views, synonyms, stored procedures etc.
- ② Add a new column for the existing table
- ③ Remove the column from the table.
- ④ Drop the table or database etc.
- ⑤ modify the database object.

* **Data manipulation language commands (DML):-**

① These commands will work with data.

- ② Inserting the records in the table.
- ③ deleting the records from the table.
- ④ update the data within the table.

* **DQL:- Data Retrieval language commands.**

These cmd's are used to retrieve data from the database objects like tables, views, synonyms etc.

* **TCL:- Transaction control language commands.**

These cmd's are used to perform transact.

* **DCL:- Data control language commands.)**

These commands are used to granting and denying the permission.

(1) * **Syntax:-** For creating database.

[create database database name ;]

e.g.: create database Satya

(2) * How to view the created database

Goto view → + object Explorer

+ servername (computer name)

+ databases

+ satya .

④ query to display the information of the database

syntax: `SP_HELPDB database name`

ex:- `SP_HELPDB SATYA`

when we execute the above query it will display the information in the form of database name, id, time, owner, ...

⑤ write a query to rename the database.

syntax: `SP_RENAME DATABASE database name`

ex:- `SP_RENAMEDB 'olddatabase name', 'newdatabase name'`

ex:- `SP_RENAMEDB 'SATYA', 'SATHYADB' (a)`

`SP_RENAMEDB SATYA, SATHYADB`.

⑥ write a q. to drop the database.

syntax: `SP_DROP DB_NAME`

`DROP DATABASE database name`

ex:- `DROP DATABASE SATYA`

Operators:-

* An operator is used to perform operation on two or more operands.

expression:- is the combination of two (2) more operators ex: $2+3*6$

Different operators in SQL Server.

① Arithmetic operators → +, -, *, / ; %

② Logical operators → and, or

③ Relational operators → <, >, <=, >=, =, !=

④ String operators → like, notlike.

⑤ Range operators → between, not between

* Arithmetic operators:- are used to perform arithmetic additions, subtraction, multiplication, division.
when ever we are solving an expression
we have to solve based on priority of
operators.

1. st priority $\rightarrow *$, $/$, $\%$.

2nd priority $\rightarrow +$, $-$

3rd priority $\rightarrow =$

\rightarrow will give quotient ex: $5/3 = 1$

\rightarrow will give remainder ex: $5 \% 3 = 2$

$$\begin{array}{r} 4 - \frac{3*5}{15} + 6 - 6/5 + 3*7 \\ \underline{-15} \\ 4 - 15 + 6 - 1 + \frac{3*7}{21} \\ \underline{-11+6-1+21} \\ \underline{-5-1+21} \\ \underline{-6+21} \\ \rightarrow 15 \end{array} \quad \begin{array}{l} 2 - 3*5 - 7/9 + 9/13 - \frac{23}{27} \\ 2 - 15 - 0 + 0 - 0 \\ (-13) \end{array}$$

* If numerator is less than denominator then the quotient is '0' and remainder the is numerator.

$$3/5 = 0 \quad 13 \% 21 = 13$$

$$7/9 = 0 \quad 3 \% 21 = 3$$

$$9/11 = 0$$

$$\textcircled{1} \ 3*3 - 3 + 3/3 - 3 \% 3 + 3 + 9$$

$$\textcircled{2} \ 5 \% 7 - 5 * 8 + 8 \% 7 * 4 - 3$$

$$\textcircled{3} \ 7 - 4 * 3 + 5 \% 8 + 8 \% 5 - 8/5 - 31$$

14.11.4 * Any thing we are inputting from keyboard is data.

Data is meaningless.

ex:- 101, anil, 20000

* What is information?

is processed data.

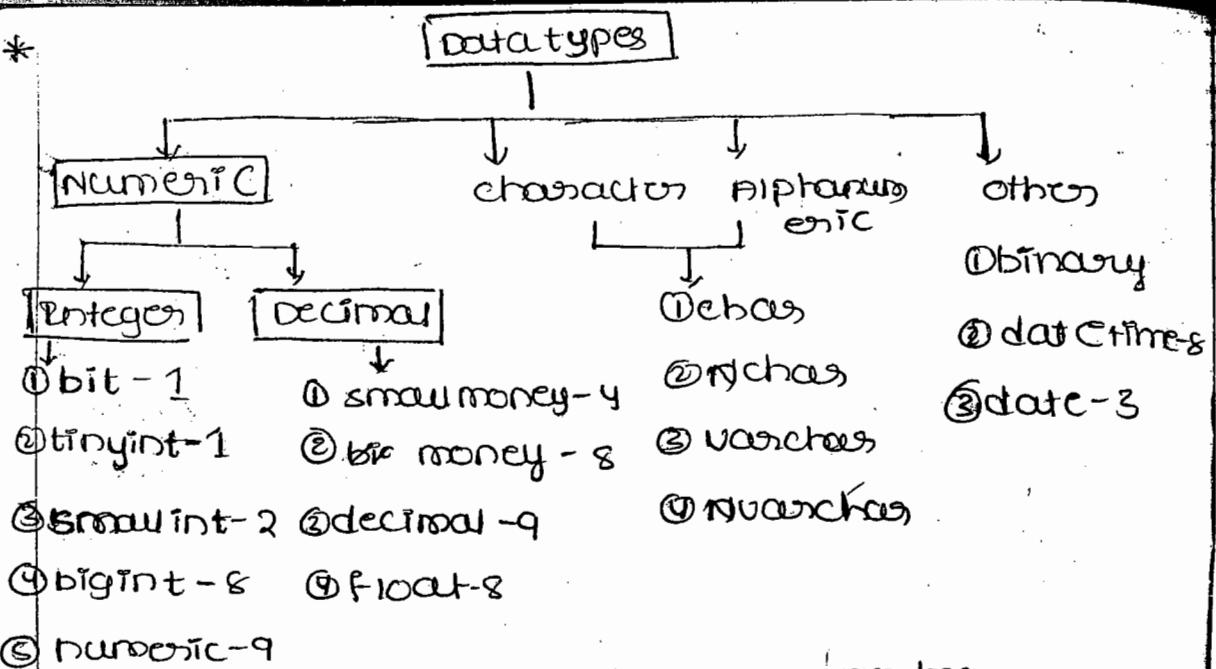
* Information is meaningful data?

ex:- Anil salary is 20000.

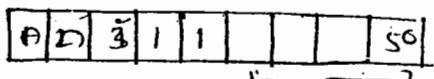
* In any programming language we will handle data because of two reasons

(1) Data will occupy less memory compared to information

(2) Easy to access the data.



* what is the difference between char, nchar, varchar, char(some size) :- The size of char is fixed.
 whenever we declare char of (50) then 50 character of size memory is allocated, if we store only 4 characters of data then the remaining 46 characters of memory is wasted.



waste memory.

varchar(some size) :- The size of varchar is not fixed (variable length)

whenever we declare varchar(50) memory is and if we store 4 char of data then varchar will allocate only 4 characters of memory so the remaining memory is not wasted.

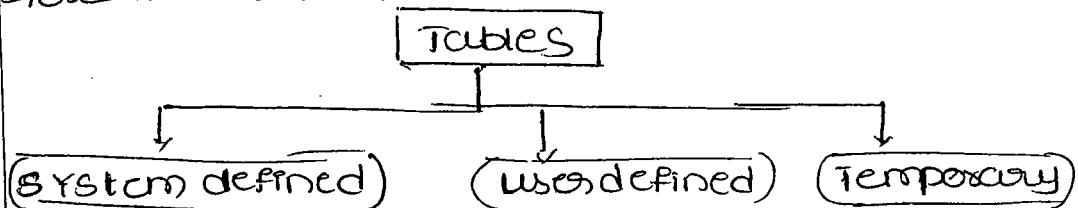
* The maximum size of charc and varchar is 8000 characters.

By using char and varchar we cannot store international languages data like chinese, Japanese, etc.

nchar(size):- The size of nchar is fixed. it is used to store any international languages data .

nuvachan(size) :- The size of nuvachan is not fixed same like vachan.
*The maximum size of nchav and nuvachan is 1000.

Table: Table is database object which is used to store the data in the form of rows & columns.



*These tables are available automatically when we install SQL Server management studio.

-The tables that was created by the user depending on the user requirement are called as user defined tables.

Create / table

~~create database table emp~~

* Create database scity a

use satya

```
use sathyam  
create table employee(eno, ift, ename, varchar(50)  
                      salary money, dob, date,  
                      emailid varchar(50),
```

* Create database student

use student

create table student(sno int, sname varchar(50),
age smallint, address varchar(50),
phone bigint)

* Ques to view the information of a table

Syntax: sp-help [datatype]

sp-help tablename

* Ques to rename the table from student to students

SP-Rename

sp-rename oldname, newname

Ex:- sp-rename students, student.

* Ques to add a new column with name marks in student table.

Syntax to add a new column:-

alter table tablename add columnname datatype

Ex:- alter table students add marks int

Syntax to remove table:-

alter table tablename drop [columnname]

alter table students drop column marks

* Ques to remove the table from the database.

Syntax[drop table tablename]

Ex: drop table students

* Ques to rename the column?

Syntax: [sp-rename 'tablename.old column name', 'new column name']

Ex:- sp-rename 'employee.empno', 'empno'

* Ques what is the difference between truncate and drop?

Truncate and drop are both ddl commands
truncate will remove data table data but not structure.

Drop will remove table data but along with the table structure.

sy:- truncate table tablename

truncate table employee

sy:- drop table tablename

drop table employee.

Q) can we insert the record after truncating?
yes

can we insert the record after dropping the table? No

18-09-14

① DMI commands:- Data manipulation language commands.

These commands are used to

① insert the records in the table

② to delete the records from the table.

③ to update the records within the table.

insert command:-

Syntax 1: Insert into tablename values (val1, val2...)

ex:- create table employee (eno int, ename string,

* insert into employee nochar(50), salary money)
 values (101, 'ani'), 23000)

② Syntax 2: {insert into tablename (col1, col2...)
 values (value1, value2)}

Ex:-

Note:- If we want to insert specific records in the table use table & use syntax -2

③ Syntax 3:- insert into tablename (column1, column2...)

select
values (valuelist)

union all

select
~~values~~ (valuelist)

union all

values (valuelist)

select (valuelist).

ex:- insert into employee (eno, ename, salary)

select 1, 'aaa', 20000

union all

select 2, 'bbb', 20000

union all

select 3, 'ccc', 20000.

④ syntax 4:- insert into tablename (list)

values (value1 IS 1),

values (value1 IS 2),

values (value1 IS 3);

ex:- insert into employee (eno, ename, salary)

values(1, 'abhi', 20000),

values(2, 'sunil', 20000).

* Delete Command: This command is used to delete a specific record or all the records from the table.

Difference b/w delete and truncate:-

delete

- ① DML command
- ② Delete the record one by one.
- ③ delete is slow
- ④ we can delete specific record from the table
- ⑤ where condition will work with delete

truncate

- ① DDL command.
- ② Truncate will delete all the records at a time.
- ③ Truncate is fast
- ④ By using truncate we can't delete specific record from the table
- ⑤ where condition will not work with truncate.

- * If we want to delete a specific record better to use delete. If we want to delete all the records from the table better to use truncate.
- ④ WAP to delete all the records from employee.
Syntax: delete from tablename.
Note: If we want to delete a specific record from the table we have to use where condition.
- ⑤ WAP to delete employee details whose employee number is 101?
Syntax: delete from ^{employee}(tablename) where empno = 101
- ⑥ WAP to delete employee details whose employee number ~~whose~~ is in IF where dname = it.
- ⑦ update command: This command is used to update the data within the table.
Syn: update tablename set columnname = value where condition.
- ⑧ WAP to update cmpsalary to 25000 whose eno is 10?
update employee set salary = ²⁵⁰⁰⁰ where eno = 101
- ⑨ WAP to update ename to anilkumar, salary to 30000 whose eno is 101?
update employee set ename = "anilkumar", salary = 30000
where eno = 101.

break

19-11-14 DQL Commands:- By using select command we can retrieve all the records or specific records from the table.

* is used to retrieve all the records from the table.

Syntax to retrieve records: `Select * From table name`

Q) wq to display employee details from employee table.

Ans: `Select * From employee`, `Select * From employee Where i=1`
when we execute

the above query it will first check the condition
 $i=1$ as the condition is true all the records
are displayed.

* Syntax to display specific records from the table?

`Select column1, column2, ... From table name`

* wq to display cname and salary from employee table?

`Select cname, salary From employee`.

Operators:-

① + addit-

② - subtract

③ * - multiplication.

④ / - quotient
remainders.

⑤ %

Q) Create a table with name student (id1)

columns, sno, sname, m₁, m₂, m₃.

`Create table student (sno int, sname varchar(20),
m1 int, m2 int, m3 int)`

(insert into student values ('101', 'anil', 70, 70, 70))

`Select * From student`)

(select *, ~~m1+m2+m3~~ as 'Total' from student)

Q) write a query to display the total marks of anil?

→ (select m₁+m₂+m₃ as 'Total' from student where sname = 'anil')

Q) waq to display student details along with total marks and percentage of marks.

→ (select *, m₁+m₂+m₃ as 'Total', (m₁+m₂+m₃)/3.0 as '%' from student.)

Q) create a table with name book with columns bid, bname, qty, price.

→ create table book (bid int, bname varchar(20), qty int, price money)

Q) to display the book details along with total bill

→ select *, (qty * price) as 'total bill' from book

Q) to display the price of C#.net.

→ select price from book where bname = 'C#.net'.

Q) create a table with columns eno, ename and bsal

→ create table employee(eno int, ename varchar(20), bsal money)

Q) to display employee to display da, hra, total sal

{ select *, (0.2 * bsal) as 'da' from employee

select *, (0.4 * bsal) as 'hra' from employee

select *, (bsal + (0.2 * bsal) + (0.4 * bsal)) from employee.

select *, (0.2 * bsa) as 'Da', 0.4 * bsa as 'Hra',
(bsa + (0.2 * bsa)) + (0.4 * bsa) as
'Totalsa' from employee.

④ write a Q to display employee details
whose salary > 20000.

select * from employee where salary > 20000.

Comparison operators

< less than

> greater than

>= greater than or equal

<= less than or equal.

!= not equal

= equal

⑤ Q to display emp details whose salary
!= 3000.

select * from employee where salary != 3000.

⑥ Q to display emp names whose salary
greater than 25000?

select * from employee where salary >= 25000

⑦ Q to display employee details whose
age is > 23?

select * from employee where age > 23.

Logical operators: and, or

These operators are used to compare
two or more conditions.

G ₁	G ₂	and	or
T	T	T	F
T	F	F	T
F	T	F	T

		eno	ename	salary	dname
1)	query to display Emp details who are working in it department and whose age > 20	101	anil	20000	IT 22
		102	sunil	23000	IT 23
		103	ajay	30000	IT 22
		104	akil	20000	IT 23

Q. → select * from employee where dname = 'IT'
and age > 20

- ① To display employee whose age is less than 23 and age < 23 salary > 20000?
- select * from employee where age < 23 and salary > 20000 o/p: ajay
- ② query to display employee details working in it department or whose age is > 23
- select * from employee where dname = 'IT' or age > 22 o/p: all the data.

③ Range Operators:- between and not between
query to display employee details whose age between 18 and 25.

- select * from employee where age between 18 and 25
- query to display employee details whose age not between 18 and 25.
- select * from employee where age not between 18 and 25.
- query to display employee details whose salary range between 20000 & 23000
- select * from employee where salary between 20000 and 23000

* String Operators :- like, & not like

like operator is used to display the text in a specific pattern.

a% starts with a

%a ends with a

an% starts with an

%a% in between string search for a

- ② to display employee details whose name starts with a?

select * from employee where ename like
'a%'

- ③ to display employee details whose name starts with a and whose salary > 25000

→ select * from employee where ename like
'a%' and salary > 25000

- ④ to display emp details whose name not starts with s and whose salary > 23000

select * from employee where ename not like 's%' and salary > 23000.

Constraints :- constraints are used to apply condition on database objects like tables.

Different types of constraints in SQL are

1. null constraints

2. not null constraints

3. primary key constraints

4. unique key constraints

5. foreign key constraints

6. composite primary constraints

7. default constraints

8. candidate key constraint

9. checked constraint

① NULL Constraint :- we can apply the constraint at the time of creating the table (or) after creating the table.

when we apply null constraint on the columns
then that column will allow null values.

→ By default when we are creating a table then null constraint will be applied on every column

Syntax:- for applying new constraint at the time of creating the table.

syn:-execute table tablename (colname datatype)

Not null Constraint:-

null, column name
datatype)

not null constraint will not accept null values when ever we are applying not null constraint on a column then that column will not accept null values.

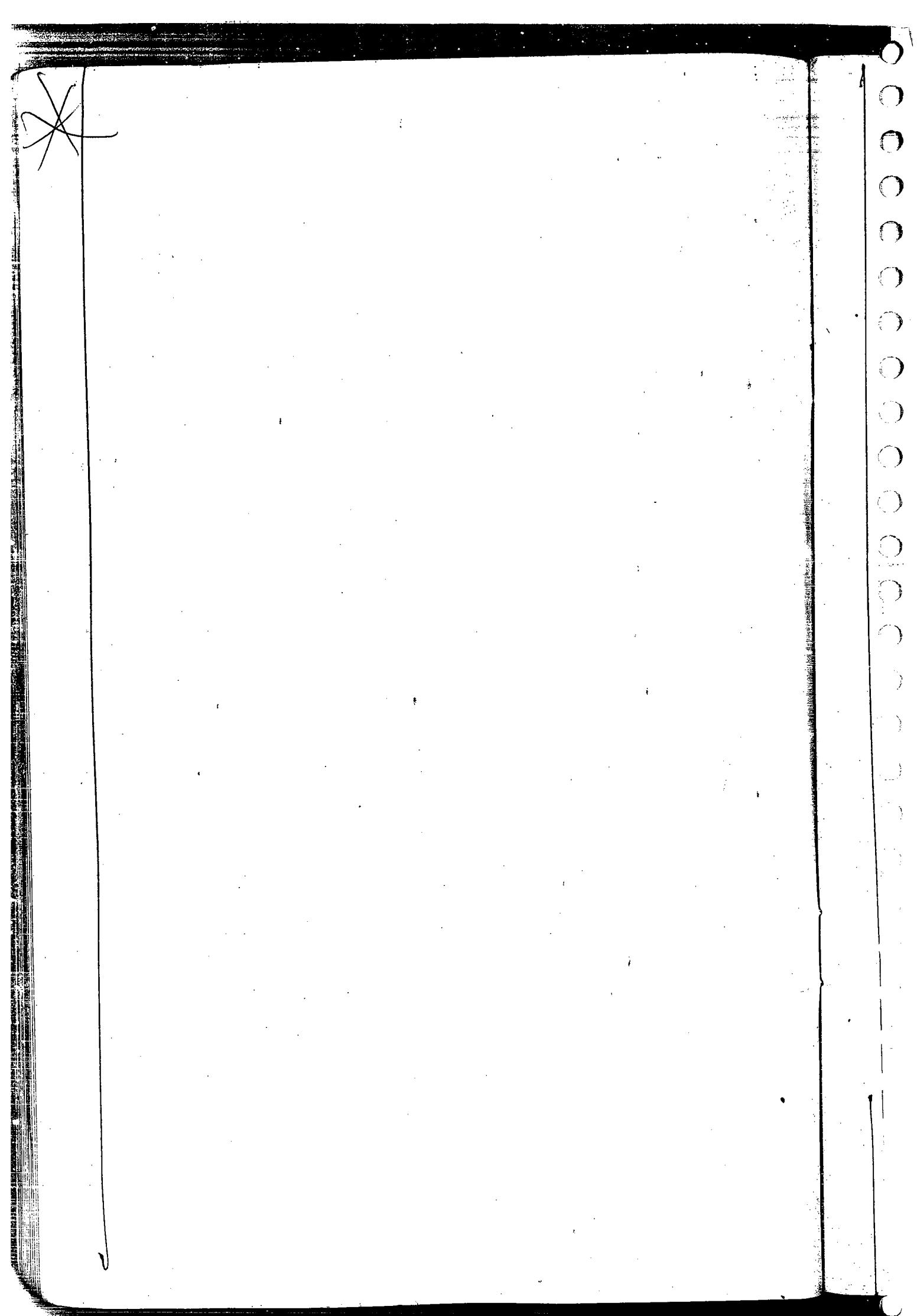
In the above table ~~one~~^{only one} column will not accept null values.

end	enamec	salary
101	null	null ✓
perf	anil	2000.00 x

Want to change the datatype of the columns from int to varchar(20), salary(money).

```
alter table emp alter column eno varchar(20)
```

Syntax: to apply NOT NULL constraints after creating the table.



→ Syntax to apply null constraint after creating the table

```
alter table tablename alter column columnname datatype  
null
```

primary key constraint :-

- 1) primary key constraint will not accept null values.
- 2) primary key constraint will not accept duplicate values.
- 3) we cannot apply more than one primary key constraint on a single table.
- #) primary key constraint will have constraint name.
- ⑤ why primary key constraint will not accept null values?

→ whenever we apply primary key constraint on a column in a table then automatically not null constraint will be applied on the primary key column and that not null constraint will not allow null values.

→ we can apply primary key constraint at the time of creating the table or after creating the table.

Syntax :- To apply primary key constraint at the time of creating the table.

Create table tablename (columnname datatype
primary key, columnname datatype)

Ex:-

create table e1 (eno int primary key,
ename varchar(20), salary money)

Note:-

When ever we apply primary key constraint
on a column the system will generate constraintname

Q) What is use of constraintname?

A) constraintname is the name for primary key
constraint

Q) What is use of constraintname?

A) By using constraintname we can drop the
constraint.

Syntax-2 :- To apply primary key constraint at the
time of creating the table.

Create table tablename (columnname datatype
constraint constraintname primary key,
columnname datatype)

Ex:- create table e2 (eno int constraint pk_e2_ano
primary key, ename varchar(20))

Q) How to view the constraintname?

④ Object Explorer

+ database

+ sathya

+ tables

+ e2

+ keys

+ PK_e2_eno

* Syntax to apply primary key constraint after creating the table. 8/11/14

```
alter table tablename add primary key (colname)
```

Note:-

We cannot apply primary key constraint on a nullable column.

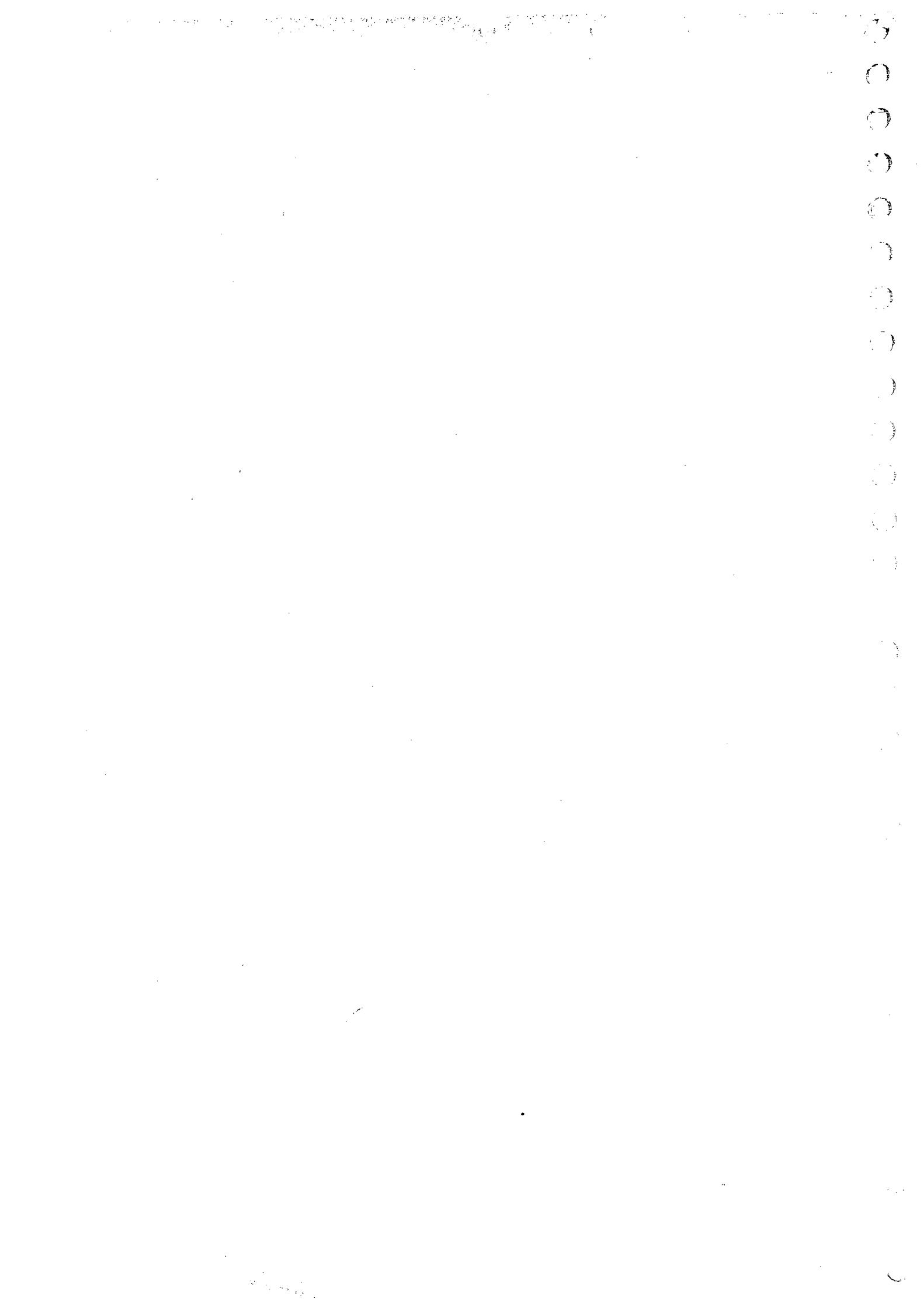
→ If we want to apply primary key constraint after creating the table we have to make the column not null.

Ex:-

```
create table emp (eno int, ename varchar(20),  
salary money)
```

```
alter table emp add primary key (eno)
```

Observation:- When we execute the above alter query it will display error message saying that cannot apply primary key constraint on nullable column. on eno so if we want to apply pkey constraint we have



* syntax to apply primary key constraints after creating the table

alter table 'tablename' add primary key (column)

Note:- we can't apply primary key constraint on a nullable column.

If we want to apply primary key constraint after creating table we have to make the columns as not null.

```
Create Table 'emp'(eno,int,ename varchar(20),  
salery float)
```

ALTER TABLE emp(eno) ADD PRIMARY KEY(eno)

Observation when we want to excute the above alter query it will display "error message" saying that can't apply p key constraint on null able column i.e. on eno.

so if we want to apply primary constraint we have to make π^a are not null.

alter table emp alter column eng int not null

alter table emp add primary(eno);

Syntax-2 To apply primary key constraint after creating the table.

```
alter table 'tablename' add constraint constraintname primary key (colname)
```

cutter table ~~empty~~ table

ex: alter table emp add constraint

pk - emp0 - eno primary (key(n))

Syntax to drop the primary key constraint:

ALTER TABLE tablename DROP constraint
constraintname.

unique key constraints:

1. unique key constraints will not accept duplicate values
 2. Unique key constraint will accept null values
 - 3 we can apply more than one unique key constraint on a single table
 4. unique key constraint will not accept a null values because it will not accept duplicate values.

Syntax: For unique key constraint at the time of creating the table

```
alter create table tablename (colname datatype  
unique, colname datatype)
```

Ex:- Create table ei (eno int primary key, ename varchar(20), phone no bigint unique)

Syntax(2)- For applying unique key constraints.

create table tablename (columnname datatype, columnname datatype, ...)

```
constraint constraint_name, column_name data  
create table e2(eno int constraint uq_e2_eno  
unique, ename varchar(20))
```

Syntax-1 to apply unique key constraints after creating the table:-

ex- alter table ~~emp~~ add unique (ename)

Syntax-2: alter table tablename add constraint
constraintname unique (columnname)

syntax - to drop unique key constraint :-

alter table tablename drop constraint constraint_name

alter table tablename drop constraint constraints_name;

④ unique + not null constraint (candidate key):

- ① it will not accept duplicate values similarly
- ② unique + not ^{constraint} will not accept null values
- ③ we can apply more than one unique + not null constraint on a single table.
- ④ unique + not null constraint will have constraint name
ex:- create table emp(eno int not null unique,

		ename	varchar(20)		
	null	notnull	pkey	ukey	u+notnull
① duplicate value	✓	✓	✗	✗	✗
② null values	✓	✗	✗	✓	✗
③ no of keys	>1	>1	1	>1	>1
④ constraints name	✗	✗	✓	✓	✓

* Default Constraints: It is used to insert default value instead of null value.

Syn:- Create table tablename (columnname datatype,
default(defaultvalue));

22-11-12
* checked Constraint: It is used to apply some restrictions on the column data.

Syntax: at the time of creating the table

create table tablename (columnname datatype,
check (condition), columnname datatype)

create a table name with name employee
with eno, ename, salary apply checked
constraint for eno column condition
is eno > 100.

* create table emp (eno int check(eno > 100),
ename varchar(20), salary
money)

Syntax: create table tablename (columnname

datatype constraint constraintname check(condition)

- * create a table with name person, with column person id, person name, age apply checked constraint for age column based on condition age must be greater than 18 age must be less than 25.

* create table person(pid int, pname varchar(20)
int constraint chk_person
age check(age > 18 and age < 25))

Syntax: for applying checked constraint after creating the table.

alter table tablename add constraint check(condition)

ex:- create a table with name employee with columnseno, ename and salary apply checked constraint for salary > 8000 and salary < 1200.

* execute alter table emp add constraint check(salary > 8000
& salary < 12000)

Syntax 2:- alter table tablename add constraint
constraintname 'check(condition)' :

Syntax to drop the checked constraint:

alter table tablename drop constraint constraint
name

Composite primary key constraint:-

Applying primary key constraint for
the combination of column f is called composite
primary

- ① CPK will not accept null values
- ② CPK will not accept duplicate values
- ③ we can't apply cpk constraint for more than one column.
- ④ CPK will have constraintname.
- ⑤

④ RPK -

employee

eno	ename	address
101	Anil	Hyd
102	Bunil	Hyd
103	Akil	
104	Ajay	Vijayawada
105		

cid	ename
1	C
2	C++
3	Java
4	Net
5	MVC

Instructors

eno	cid
101	1
101	2
101	4
102	1
102	3
103	null
null	null

execute table employee (eno int primary key,
ename varchar(20),
address varchar(30))

Create table ^{com}courses(cid int primary key,
course varchar(10))

Syntax for applying composite primary
key constraint.

create table tablename (colname1 datatype,
colname2 datatype,
primary key (col1, col2))

ex: create table faculty (eno int, cid int
primary key (eno, cid))

syntax:

create table tablename (colname1 datatype,
colname2 datatype,
constraint constraint name,
primary key (col1, col2))

syntax: to add CPK constraint after creating
the table alter table tablename add primary
key (col1, col2).

syntax to drop the constraint

alter table tablename drop constraint
constraint name.

Normalization: Normalization is a database design
technique which is used to reduce redundancy
and dependency of data.

<u>Item types</u>	<u>Items</u>	<u>Qty</u>	<u>Price</u>
pizza	VP, CP, ChP	10, 20, 30	60, 90, 120
burgers	VB, CB	15, 20	75, 90
cooldrinks	SPRITE, TUMS, C	30, 20, 30	22, 23, 18
Cakes	PASTRIES, BS, C	20, 30, 30	12, 20, 30
puffs	VP, CP	40, 50	20, 30

The above data is unnormalized form. To make the above table as normalized we have to apply 3-normal forms

1-normal forms:-

1 rule:- single cell consists of single value.

Item type	Items	qty	price
pizza	vp	10	60
pizza	cp	20	90
pizza	chp	30	120
Burgers	vb	15	70
burgers	cb	20	90
cool drink	sprite	30	22

2-rule:- each record must be unique.

A table is said to be in first normal form if it obeys first & second rule.

2-Normal Form:

(1) A table is said to be in second normal form if it obeys below rules

1-rule:- Table must be in first normal form.

In second normal form identify the key attribute and non key attribute.

→ Every non key attribute must fully dependent on key attribute.

→ If any non key attribute is not depending on key attribute remove that column and throw in a separate table.

→ How to identify key attri

The maximum no of columns must depend on another column.

In above table qty and price fully dependent on items. So items is key attributes.

Item type is not depending on items.

Remove item types and place in a separate table.

<u>item master</u>		<u>item</u>			
tid	type	ino	lname	dty	price tid
t1	pizza	1	mpizza	10	60 t1
t2	burger	2	mpizza	20	90 t1
t3	coldrink	3	mburger	15	70 t2
t4	puff	4	mburger	20	90 t2
t5	cake	5	sprite	30	22 t3
		6	coke	20	20 t3
		7	mpuff	10	12 t4
		8	pastry	10	22 t5

* In second normal form we have to apply the primary keys and foreign keys.

In item master table tid \rightarrow primary key.

In items table - ino \rightarrow primary key

tid \rightarrow foreign key

* The primary key of one table visiting to another table will become, foreign key.

@ What is foreign key :- is used to establish the relationship between two or more columns.

foreign key constraint

- ① It is used to apply relationship between 2 or more table.
- ② Foreign key constraint will accept duplicate values.
- ③ Foreign key constraint will accept null values.
- ④ We can apply more than one foreign key constraint on a single table.
- ⑤ Foreign key constraint will have constraint name.
- ⑥ The primary key of one table will become foreign key in another table.
- ⑦ We can't insert the value in the foreign key if the value is not available in primary key column.
- ⑧ We can't delete primary key column until and unless we delete foreign key column.
- ⑨ We can't delete the value from primary key column until we delete the same value in foreign key column.

Syntax to create foreign key constraint:-
at the time of creating table

* Create table tablename (columns datatype
foreign key references pkeytablename
(pkeycolumn))

- ⑩ Create a table with name employee with columns eno, ename, salary, and apply pkey for eno?

Create table emp(eno int primary key,
ename varchar(20),
salary money)

create a table with name dept with columns dno, dname, eno, apply foreign key constraint for eno column?

* create table dept(dno int, dname varchar(20), eno int, foreign key references emp(eno))

* Note: The datatype of primary key column and foreign key column both must be same.

Syntax ② create table tablename (columnname datatype constraint constraintname foreign key references pkey tablename (pkey columnname))

Syntax to apply foreign key constraint after creating the table

alter table tablename add foreign key(columnname) references pkey^{table} name (pkeycolname)

alter table emp add foreign key(eno) references de emp(eno).

* Syntax to drop foreign key constraint
alter table table drop constraint constraintname.

① Create a table with name bus columns & busno, svno, travelname, starttime, reachtime, fromplace, toplace, capacity and apply primary key for svno.

② Create a table with name destination with columns & did, dname, fare and apply pkey constraint for did.

- ④ create a table with name pickuppoint with columns pid, pname and apply pkey for pid?
- ⑤ create a table with name faredestination with columns svno and did apply foreign key for svno, did and pkey for (svno, did).

execute a table faredestination
(svno int foreign key references bus(svno), did int foreign key references destination(did); primary key(svno, did))

- ⑥ create a table with name buspickuppoint with columns svno, pid . apply foreign key for svno, pid and pkey for (svno, pid)?

joins:- are used to retrieve the data from than one Table.

Different types of joins:-

① inner join

② outer join

- ① left outer join
- ② right outer join
- ③ full outer join

④ cross join

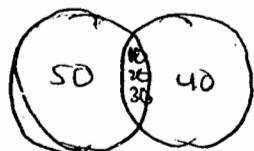
⑤ self join

⑥ equijoin

⑦ non-equijoin

* If we want to work with joins min two tables must be available. and a common column between two tables.

* Inner join:- it is used to display the matching records from both the tables



emp				
eno	ename	sal	bno	
101	anil	20000	10	
102	sunil	30000	20	
103	john	20000	10	
104	sagar	15000	30	
105	ram	30000	20	
106	Jacob	20000	50	

department	
dno	dname
10	it
20	maths
30	physics
40	marketing

Syntax 1: select tablename1.col1, tablename2.col2...
 from tablename1 inner join tablename2
 on condition.

Q. write a q to display emp details along with department name.

select emp.eno, emp.ename, emp.sal, emp.dno,
 dept.dname
from
 emp inner join dept
on emp.dno = dept.dno.

Q. write to display ename, sal, department name
 from both the tables.

select emp.ename, emp.sal, dept.dname
from emp inner join dept
on emp.dno = dept.dno

Q. write to display employee details?

select * from emp.

Q. write to display employee det & ename, sal
 who are working in it

select emp.ename, emp.sal,
 from emp inner join dept
 on emp.dno = dept.dno where dept.dname = 'it'

observation:- whenever we execute the above query the inner join will match join matching records in the table and then where condition is applied on table and ^{display} ename, and salary will be displayed.

④ write a qu. to display employee details along with department name whose salary is > 2000.

select emp.empno, emp.ename, emp.sal, emp.dno
dept

select emp.* , dept.dname
from emp inner join dept
on emp.dno = dept.dno
where emp.salary > 20000

⑤ write to display employee details along with department name whose salary range between 20000 - 25000.

select emp.* , dept.dname
from emp inner join dept
on emp.dno = dept.dno
where emp.salary between 20000 and 25000

⑥ write to display emp details along with dept name whose name starts with A.

select emp*, dept.dname
from emp inner join dept
on emp.dno = dept.dno
where emp.ename ^{like} 'A%'
where ename like

'ay.'

- ⑧ create two tables with name state and city

state

sid	sname	sd
s1	AP	1
s2	Telangana	2
s3	TN	3
s4	Kerala	4
s5	Karnataka	5

city

ename	sid
Hyd	S2
Mysore	S3
Vizag	S1
Vijaya	S1
Chennai	S3
Madurai	S3
Banglore	S5
Kochi	S4

- ⑨ query to display city name and state names

select city.ename, state.sname from state inner join city on state.sid = city.sid

- ⑩ query to display city names belong to AP

select city.ename from state inner join city on state.sid = city.sid
where state.sname = 'AP'

- ⑪ query to display the state name of Kochi

select state.sname from state, city
inner join city
on state.sid = city.sid where city.ename
= 'Kochi'.

- ⑫ query to display the city name belong to telangana whose name starts with C

select city.ename from state inner join city on state.sid = city.sid, where
city.ename like 'H%' and state.sname = 'Telangana'

- 27-11-24
⑥ Create a table with name bus with columns svno, tname, fromplace, toplace, starttime, reache time, capacity?

Create table bus(svno, varchar(20), primary key, tname varchar(20), from place varchar(20) → reache time varchar(20), capacity int).

- ⑦ Create a table with name destination with columns did, dname, fare?
- * Create table destination(did varchar(20)) primary key, dname varchar(20), fare money
- ⑧ Create a table with name pickuppoint with columns pid, pname?

* Create table pick point(pid varchar(20)) primary key, pname varchar(20))

- ⑨ Create a table with name busdestination with columns svno and did and apply foreign key for svno, did and primary key for svno, did?

* Create table busdestination(svno, varchar(20)) Foreign key references bus(svno), did varchar(20) Foreign key references destination(did), primary key(svno, did))

- ⑩ Create a table with name buspickup with columns svno, pid apply foreign key for svno, pid and pkey for svno, pid?

Create table buspickup(pid, svno, varchar(20)) Foreign key references bus(svno), pid varchar(20) Foreign key references

pickup point (pid), primary key (svno, pid)):

- ① write a q to display bus table
select * from bus
- ② write a q to display travel agency names
select distinct (tname) from bus.
distinct is used to display the unique values
that exist within the column.
- ③ write a q to display the travel agency name
starts with 'K'.
select distinct (tname) from bus where tname
like 'K%' bus
- ④ way to display the timings of the board
travel agency name starting from hyd to
vizag?
select tname, startime, reachime from
bus where fromplace = 'Hyderabad' and
toplace = 'vizag'.
- ⑤ way to display the fair, of the bus
source no of the bus travelling from Hyderabad
to Rajamundry?
select b.svno, b.startime, b.reachime, bd.fare
from destination d inner join busdestination bd
~~# inner join b on bd.did = bd.did~~ inner join
bus b on bd.svno = b.svno where
b.fromplace = 'Hyderabad' and d.name = 'Rajam
undry!'
- ⑥ way to display the capacity of the bus
travelling to chennai?
- ⑦ way to display the fare of the bus which is
travelling to vijayawada)
- ⑧ way to display the destination points
of sv12

- (16) query to display the pick points of kaveri
 travels from Hyderabad to vizag.
- (17) query to display start time and reach time
 and fare of the bus travelling from Hyderabad
 to nellore?

employee (e)

eno	ename	salary
101	anil	20000
102	sunil	22000
103	john	23000
104	djay	30000

dept (d) ①

dno	dname
10	IT
20	maths
30	physics

empdept (ed)

eno	dno
101	10
102	20
103	10
104	10

* writing to display details along with dname,

select e.eno, e.ename, &.salary, d.dname

from employee inner join empdept ed

on e.eno = ed.eno

inner join dept d

on d.dno = ed.dno

② display emp details working in IT.

select e.*

from dept d inner join empdept ed
on d.dno = ed.dno
inner join emp e
on e.empno = ed.empno. where d.dname = 'IT'

③ display the dname of anil

select d.dname

from emp e inner join empdept ed
on e.empno = ed.empno
inner join dept d
on d.dno = ed.dno.
where e.ename = 'Anil'.

④ display the empnames who are working
in 'IT' department whose names start
with 'A'

select e.ename

from dept d inner join empdept ed
on d.dno = ed.dno
inner join emp e
on emp.e.empno = e.empno
where e.ename = 'A%' ^{A%} as d.dname = 'IT'

⑤ display emp details working in maths

salary > 2000.

select e.*

from dept d inner join empdept ed
on d.dno = ed.dno.
inner join emp e
on emp.e.empno = ed.empno = e.empno

where d.dname = 'maths' & e.sal \geq 20000
 → display emp details working in (it) whose salary range between 20,000 and 25000.

select e.*

from dept d innerjoin empdept ed

on d.dno = ed.dno.

inner join emp e

on e.empno = ed.empno

where e.sal between 20000 and 25000
 & d.dname = 'it'

* Left Outerjoin :- It is used to display all the records from left table and matching records from right table and if there are no matching records display null values.

(E)

eno	ename	sal
101	Anil	20000
102	Sunil	23000
103	John	24000



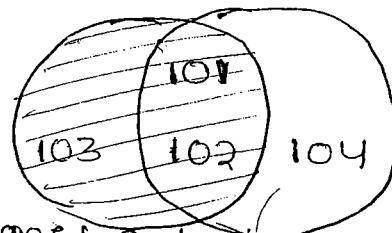
(D)

dno	dname	eno
10	it	101
20	maths	102
30	physics	104

Output

eno	ename	sal	dno	dname	eno
101	Anil	20000	10	it	101
102	Sunil	23000	20	maths	102
103	John	24000	30	physics	null

Partly
done

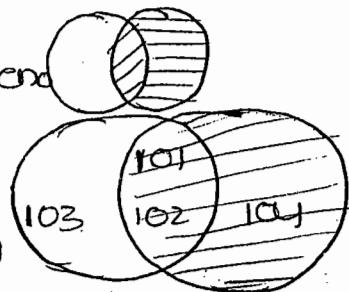


select e.*, d.* from emp e left outer join dept d on e.empno = d.dno.

Right outer join :- It is used to display all the records from right table and matching records from left table if there are no matching records in left table and display null values.

Output

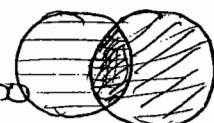
eno	ename	sal	dno	dname	eno
101	Anil	20000	10	IT	101
102	Sunil	23000	20	MATHS	102
new	null	null	30	PHY	104



select e.* , d.* from emp e right outer join dept d on e.dno = d.dno

* full outer join :- It is used to display all the records from both left table and right table and if there are no matching records in both the table display null values.

eno	ename	sal	dno	dname	eno
101	Anil	20000	10	IT	101
102	Sunil	23000	20	MATHS	102
103	John	24000	Null	Null	Null
Null	Null	Null	30	PHY	104



select e.* , d.* from emp e full join dept d on e.dno = d.dno .

~~Q11.11~~ ~~write Q/R~~

is operator is used to display null values from the table.

eno	ename	sal
101	Anil	Null
102	Sunil	20000

select * from

query to display the salary of Anil?

select sal from emp where sal = null

* select sal from emp where sal is null

table 1

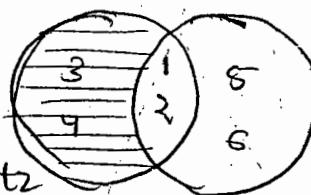
id	Name
1	one
2	two
3	three
4	four

only left record

table 2

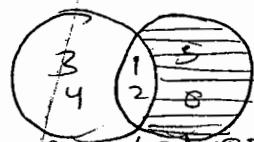
id	Name
1	one
2	two
3	five
4	six

① select t1.*
from table1 t1 inner join
on t1.id = t2.id
where t2.id is null



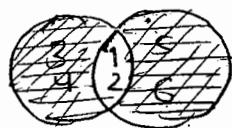
② only right records.

select t2.*
from table1 t1 outer right join table2 t2
on t1.id = t2.id where t2.id is null.



③ only left & right records

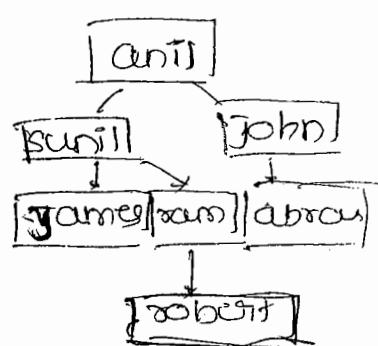
select t1.* , t2.*
from table1 t1 full join table2 t2
on t1.id = t2.id
where t1.id is null || t2.id is null.



~~Self JOIN:- means joining the table by itslef
one is called self JOIN~~

Self

eno	ename	sal	mid
1	anil	20000	null
2	sunil	23000	01
3	john	25000	2
4	james	26000	3
5	ram	30000	2
6	abca	25000	3
7	robert	30000	5



write a query to display emp name and manager name.

while working in selfjoin we have to consider two alias name e, m
e \rightarrow employee

m \rightarrow manager

mid in employee table is manager in m table i.e. ename in e table is employee ename in m table in manager.

e.mid = e.empno	employee	manager
e.mid = m.empno	e.ename	m.ename
1	sunil	anil
1	john	anil
2	james	sunil
2	sam	sunil
3	abrar	john
5	robert	sam.

[Select e.ename as 'employee', m.ename as 'manager' from employee join employee m on e.mid = m.empno.]

1-12-14

Cross Join: Cross join is the cartesian product of no of rows from left table with no of rows from right table if there are m rows in left table and n rows in right table then the cartesian product is m*n rows i.e. each row of left table will join with each row of right table.

emp			dept		
eno	ename	sal	dno	dname	eno
1	anil	20000	10	IT	1
2	sunil	30000	10	maths	2

out put

eno	ename	sal	dno	dname	eno
1	anil	20000	10	IT	1
1	anil	20000	20	maths	2
1	anil	20000	30	physics	3
2	sunil	30000	10	IT	1
2	sunil	30000	20	maths	2
2	sunil	30000	30	physics	3

* EqwJoin: It is used to display the data for more than one table by using equal operations.

⑥ Ques. To display emp details along with dno, dname from both emp & dept.

select e.* , d.dno, d.dname from emp e, dept
where e.eno = d.dno .

(equal operator)

8

Notes in equi join innerjoin replaced with , and on condition is replaced with where

⑥ NON EQUI JOIN: It is used to retrieve the data from more than one table without using equal operator.

We can use any other operators except equal ($=$) operator.

emp			salgrade		
eno	ename	salary	lowsal	highsal	grade
101	anil	20000	25000	30000	A
102	sunil	23000	22000	24000	B
103	john	24000	19000	20000	C
104	James	30000			

Write a query to display emp details along with grade.

select e.* , g.grade from emp e, salgrades
where e.sal between s.low and s.hsal

* Ques, to display student details along grade.

sno	sname	marks	(grade)	
			inmark	hmark
1	anil	70	75	100
2	sunil	65	60	74
3	john	90	80	89

select s.* , g.grade from student s, grades g
where s.sno between g.grade.g.

function:- is a sub program which is used to return only one value.

The return type of function may be int, float, varchar, money.

Function are of two types:-

① User defined function

② Predefined function

single row fun

multi row fun

① Single row function

① Mathematical

② String

③ Date time

④ Ranking

② Multi row function

Aggregate Functions

① sum()

② max()

③ min()

④ avg()

⑤ count()

→ Function will always take from user and it will process and return only one value of any data type.

→ Functions are of two types.

User defined function:- The function that was created depending on the user requirement is called as user defined function

→ Predefined function:- The function that was created by Microsoft are called predefined.

single row function:- The function which will process single row at a time and return only one value is called as single row function.

① Multi row function: The function that will process on multiple rows and return only one value.

21-10-14
② Mathematical function: This functions will take the input as number and return a numeric value.

③ Abs(number): This function will unsigned the value of given no.

Ex:- select Abs(-5) op:- 5

④ Square sqrt(no): This function will give the square of given number.

Ex:- select sqrt(25) op:- 5

⑤ Square (no): This function will give square of the given number.

Ex:- select square(7) op:- 49

⑥ Sign(num): will return 1 if number is positive. will return -1 if given number is negative.

* will return 0 if given number is zero.

Select sign(3) op:- 1

Select sign(-3) op:- -1

Select sign(0) op:- 0

⑦ Power (num): This function will give the power of given number.

Ex:- select power(2,3) op:- 8

* String functions: The functions will take input as string.

⑧ Length(string): This function is used to get the length of string.

Ex:- length('sathya')

⑨ Left(string,no): This function is used to get no of characters from left side of string.

Ex:- Select left(sathyatech,6) op:- sathya

⑩ Right(string,no): This function is used to get no of characters from right side of string.

Ex:- Right(string,no):- Right(sathyatech,4)

op:- tech.

④ lower(string) :- upper case to lower case.

ex:- select lower('SATHYA'); o/p: sathyA

⑤ upper(string) :- lower case to upper case character.

ex:- select upper('sathyA') op: SATHYA

⑥ ASCII(string) :- This function will get the ASCII value of the given character.

ex:- select

a - 97 - 122

select ASCII value(a) = 97.

A - 65 - 90

O - 9 48 - 57

@ G4

* Date time function:-

① Get Date() :- This function is used to display the current system date and time

ex:- select Getdate()

yy yy - mm - dd hh:mm:ss:msc

2014 - 12 - 02

14:36:49:680

ex:- select day(get date)

② waq to display the current system day?

ex:- select day(get date()), month (getdate()), year (getdate()) . = 02 - 12 - 2014

③ dateadd(datepart, no, date) :- This function

is used to add the number of days, months, or years to current system date?

waq to display the date after 15 days?

select DATEADD(dd, 15, getdate())

④ waq to dispicy the date after 2 months

select DATEADD(mm, 2, getdate())

⑤ waq to dispicy the date after 3 years

select DATEADD(YY, 3, getdate())

④ DATE DIFF (startdat, startdate, end date) :-
date part,

This function is used to find the difference between 2 dates.

⑤ wq, to find the difference b/w 2 dates

2014-12-02 to 2014-12-15

select DATEDIFF(dd, '2014-12-02', '2014-12-15')

⑥ wq, to create a table with name emp
with columns ,eno, cname, sal, doj .

eno	cname	sal	doj
101	Anil	20000	2013-04-02
102	sunil	23000	2012-03-03
103	ajay	30000	2014-04-02

⑦ wq, to display the emp details along with exp in terms of years?

select *, datediff(yy, doj, getdate()) as (exp)
from emp.

* Multirow functions:- The function which will process on multiple rows at a time and return only one value are called multi row functions or aggregate functions:

- ① sum()
 - ② avg()
 - ③ max()
 - ④ count()
 - ⑤ min()
- }

eno	ename	desig	salary	city	region	dno	dname	gender	dept
101	anil	dev	20000.00	hyd	S	10	IT	male	SE
102	sunit	dev	25000.00	hyd	S	10	IT	male	CO
103	agay	bde	15000.00	hyd	S	20	marketing	Male	IT
104	Karen	adm	300000.00	chennai	S	20	marketing	Male	SC
105	Rajal	frtr	20000.00	bang	S	30	HR	Female	CA
106	santosh	hr	3000.00	pune	N	30	HR	female	NC

① query to display the no of employees working in the company.

select count(*) from emp

② query to display the total salary for all employee
select sum(salary) from emp.

③ query to display max salary of employee
select max(salary) from emp.

④ query to display min salary of emp?
select min(salary) from emp.

⑤ query to display the max salary and min sal of employee working in the company

* select max(salary) as 'maxsal',
min(salary) as 'minsal' from emp.

⑥ query to display the avg sal of emp?
select avg(salary) from emp

⑦ display the dno of emp working in IT

select count(*) from emp where dname='IT'

⑧ display the no of males working in marketing dept?

select count(*) from emp where dname='marketing' @ se
and gender='male'.

⑨ to display max salary of emp working in IT department

- gender
male
female
region
North
South
East
West
- ⑥ ~~query~~
select max(salary) from emp where dname = 'TP'
 - ⑦ ~~query to count the no of females working in the company~~
select count(*) from emp where gender = 'Female'.
 - ⑧ ~~query to find the avg. sal. that assigned for North region & south region?~~

~~⑨ Clauses in SQL:-~~

- ① Order by clause :- It is used to display the data either in ascending order or in descending order.
Note :- By default order by clause will display the data in ascending order.

- ② ~~query to display emp details orders by ename.~~

select * from emp order by ename.

- ③ ~~query to display emp details order by ename in descending order?~~

select * from emp order by ename desc

- ④ ~~query to display emp details along with dno, dname, from both emp and dept tables order by ename?~~
select e.* , d.dno, d.dname from emp e inner join dept d on e.empno = d.empno order by e.ename.

~~⑤ Group by clause:-~~

~~writing a query to display the total salary of males and female.~~

* It is used to group the common set of values

that are available on single column.

* Group by clause will always work with aggregate functions.

- ⑥ ~~select gender, sum(salary) as 'Tsal' from emp group by gender.~~

In the above query first group by class will group the common set of column values that

are available in gender column. i.e. all males are considered as one group and all females are considered as another group and then the aggregate fun (sum(salary)) will be applied on male group and female group.

⑥ way to display the total savings based on
expenses

select (region), sum(salary)

as Tsai

from emp

group by region

④ way to display the avg salary based on region

select region, sum avg(salary)

as 'A sal

From emp

group by region

⑥ write a program to count the no of employee working in each city

→ select city, norferry count(*)

as nof emp(Total)

From Emp

group by region city

⑥ code to count the no of employee working in each department.

→ Select dept dname , count(*)

as Total

from emp

group by region name

- ④ way to display the total salary that was assigned to each department.

```
select dname, sum(salary)  
as total  
from emp  
group by regiondname
```

⑤ way to count the no of males & females working in the company.

```
select gender, count(*)  
as (no-of-employees)  
from emp  
group by regiongender
```

⑥ way to display the max salary and min sal of employee working in each city.

```
select city, max(salary) as 'maxsal', min(salary)  
as maxsalary  
as 'minsal'  
from emp  
group by regioncity
```

⑦ way to display the total sal based on dname order by dname.

```
select dname, sum(salary)  
as 'Total sal' from emp  
group by dname order by dname.
```

③ Having clause:- It is used to apply where condition on grouped data.

having clause will always work with group by clause.

* we can't apply **where** on group by clause.

* A query to display the total salary that was assigned for each department whose total salary is greater than 50,000.

select dname, sum(salary) as 'tsal' from emp & group by dname having sum(salary)>sal

(Q) (E)
sc

⑥ Ques. To display average salary of each department whose avg. sal>20000 order by dname?
'select dname, avg(salary) as avgsal
from emp group by dname having avgsal
>20000 order by dname.

(Q) (E)
fr
si

~~WIP-14~~ Subqueries :- Subquery is a select query inside another select query.

We can write maximum of 32 select commands in a single subquery

Syntax:- outer query(inner query)

While working with subquery always the inner query will gets executed first and based on the result of the inner query the next highest query will gets executed.

vol = no.
no.
diagram
e

⑦ Display the max salary of an employee?

select max(sal) from emp.

⑧ The second max salary of an employee.

select max(^{max(sal)}) from emp where salary <
(select max(sal) from emp)

In the above query the result of inner query will be executed i.e. 25000 and the result of the inner query is given to next highest query.

⑨ The third max salary of an emp

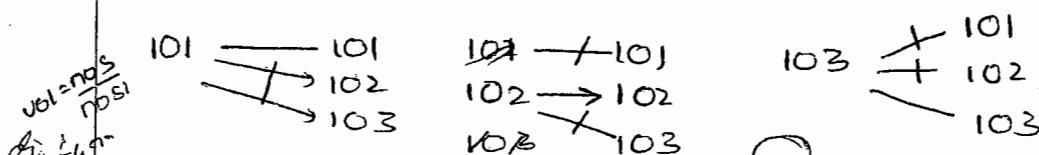
select max(salary) from emp where salary <
(select max(salary) from emp) where salary
<(select max(salary) from emp)).

~~WIP-14~~ In operator:- In operator will work like or operator

- ④ way to display emp details whose eno is 101 or 102
 select * from emp where eno in(101, 102)
- ⑤ way to display emp details who eno are not 101 and 102
 select * from emp where eno not in(101, 102)

emp			dept		
eno	ename	sal	dno	dname	eno
101	anil	20000	10	IT	101
102	sunil	23600	20	MATHS	102
103	john	26000	30	PHYSICS	103

select e.* , d.dno, d.dname from emp e inner join dept d on e.eno = d.eno



⇒ select e.* , d.dno, d.dname from emp e where e.eno in (select d.dno from dept)

where eno → (in) → (select d.dno from dept)
 101 OR 102 OR 103 } 3 condition.

- ⑥ way to display employee details working in IT department

* select (inner query → select eno where dname = 'IT')

eno	dname	eno
it	← 101	

outer query → select * from emp where eno in (select eno where dname = 'IT')

eno(emp) ← eno(dept)

101 (4) 101 (3)

② waq, to display emp details who is working in maths & sal > 20000?

③ inner query

(select eno from dept where dname is 'maths')
and salary > 20000)

outer select e.* from emp e where eno in (inner query)

④ waq, to display details who is working in IT dept & name starts with a.

① select dno from dept where dname = 'IT' and ename like 'a%')

② select * from emp where eno in (inner q)

student

sno	name	city
101	anil	hyd
102	sunil	chennai
103	ajay	hyd

course

cid	cname
1	C
2	IT
3	JNet
4	Java

enrollment

sno	cid	selct * from student where sno in (selct sno from course where cid in (selct cid from course where cname =))
101	1	
101	4	(selct sno from course where cid in (selct cid from course where cname =))
102	3	
102	2	
103	1	
103	2	
103	3	

Q9
⑨ query to display the student details who enrolled
for C.

* ①st inner query.

① id [select cid from course where cname='C')]

② [select sno from enrollment where cid in]

③ (select * sno from student where sno in)

select * from student where sno in -

Select sno from enrollment where cid in (

select cid from course where cname = 'C')))

⑩ query to display the student details along
with ename whose is staying in chennai,

① select sno from student where city = 'chennai'

② select cid from enrollment where cid in

③ select cname from course where cid in

select *, student, ename, course

~~5-12-14~~ ⑪ query to display the 3rd max salary of an
emp?

* formula to display nth max salary

select columnname from tablename alias
name1 where n = (select count(*) from alias
tablename alias name2 where condition)

ex:-

select columnname

→ select salary from emp e where s

= (select count(*) from emp f where
e.salary <= f.salary)

if (e.salary <= f.salary)

{

select count(*) from emp/f

}

e.salary <= f.salary

emp e @

emp f

→ f

sno	ename	salary
101	anil	25000.00
102	sunil	24000.00
103	kiran	23000.00
104	john	22000.00
105	frame	20000.00

sno	ename	salary
101	anil	25000.00
102	sunil	24000.00
103	kiran	23000.00
104	john	22000.00
105	frame	20000.00

if(e.salary <= f.salary)

{ 25000

≤ 25000.00(1)
≤ 24000.00(1)

| 23000.00(1)
| 22000.00(0)
| 20000.00(0)

3

| s = (count=3)
| false

to display the nth max salary we are applying self join concept we are considering emp table as two tables as @ e f

if(e.salary <= f.salary)

i.e we are checking the condition e.salary <= f.salary

if the condition is true, then count the numbers

from ftable if count is equal to s then display e.salary from e table. i.e at e.sal

e.sal = 20000

then then count = 5

condition

count

e.salary ≤ f.salary

T

1

21000 ≤ 25000

F

1

21000 ≤ 24000

F

1

21000 ≤ 23000

T

1

21000 ≤ 22000

T

1

21000 ≤ 21000

T

1

count(*) 5

* Note: the above query will not work with duplicate records.

select duplicate(salary) from emp e s =

(~~group~~ select count(distinct(salary)) from empf
where empf.empno = emp.e.empno and empf.salary = e.salary)

* we can apply subqueries on BML queries
(also also)

emp

eno	emp	sal
101	sunil	20000
102	sunil	23000
103	john	24000

emp backup

eno emp sal

* insert into emp backup
select * from emp]

* top: it is used to display the top records from the table.

① way to display the first record from the table?

select top 1 * from emp.

② way to display the first five record from emp table

select top 5 * from emp

③ way to display the maximum salary of an employee?

select top 1 salary from emp order by salary desc

salary ① → order by descending ② → top 1 ③

④ * (Select distinct and top 3 salary from emp order by salary desc.)

Step 1: all the salary from employee table

Step 2: order by class will arrange the salary

in descending order and top 3 will display the first 3 salaries.

- * want to display fifth max salary.
- ② Sub query → [select distinct top 2 salary from all the salary → display all the emp]
 - ③ order by → desc in descending order.
 - ④ distinct → which remove duplicate records
 - ⑤ top 2 → will bring top two salaries
 - ⑥ result → will store the result.
 - ⑦ order by salary → will arrange the salary in descending order.
 - (select salary from emp) result → will store the result.
 - ⑧ top 1 → will get first salary

```

    select top 1 salary from (select distinct
    top 2 salary from emp order by salary
    desc) result order by salary
  
```

- * [select top 1 salary from (select distinct top 2 salary from emp order by salary desc) result order by salary] *

Observation: when we execute above query inner query will be executed all the salary will be now from table will be displayed and the order by clause will arrange the salaries in descending order and then top 5 will be displayed the top ^{two} records and the result is stored in the result then order by salary will arrange the salary

* From class: It is used to pass the query instead of table name

ex: select * from emp

select * from (select * from emp)e

(2)

(1)

② select * from e

② select * from (select ename from emp)e
~~all the ename~~

(all the names from ename)

* Row_number(over(order by)) :- Row number function is used to assign a unique identification no for every row.

→ Row no function will always works with order by clause.

⑥ query to display emp details along rowid
order by eno.

select * from ~~emp~~(select *, ROW_NUMBER()
over(order by ename)) as 'rowid' from
emp) rowid)

* When we execute the above query inner query will be executed. Order by clause will arrange the employee details in ascending order and then row number function will assign a unique a rownum for assigned for the data.

* Way to display even records from table.

select * from (select *, ROW-NUMBER()
OVER(ORDER BY eno) AS 'rowid' FROM

emp) WHERE

Partition clause :- rowid % 2 = 0

It is used to group some common set of values
in a column and then it will apply rownumber
for the grouped data.

→ If we want to apply row-number for the
grouped data then we have to use partition
by clause

Syntax:- ROW-NUMBER() OVER(PARTITION BY
colname ORDER BY colname)

② Way to display Emp details along with row
number partition by city?

select *, ROW-NUMBER() OVER(PARTITION BY(CITY)
ORDER BY eno) AS 'rowid' FROM emp

Step 1:- Partition by(City)

Step 2:- Order By

Step 3:- Row-Numbers

In the above query partition by will
partition the common set of values that
are available in city column.

eno	city
101	hyderabad
103	hyderabad
102	hyderabad
104	chennai
105	chennai

} after partition.

and order by clause will arrange the
partitioned data in ascending order based
on eno column.

eno	city
101	Hyderabad
102	"
103	"
104	Chennai
105	Chennai

after order by

(Row-number) will assign rownumbers for the partitioned data.

eno	city	rid
101	Hyderabad	1
102	Hyderabad	2
103	Hyderabad	3
104	Chennai	1
105	Chennai	1

Q8

* query to display, ename, city, rid partitioned by ename order by ename?

select ename, city, ~~and row_number()~~ ^③
partition by ename ~~order by ename~~ ^④
as 'rowid' ^① from emp. ^②

* with clause : (CTE) common table expression

query to delete emp is a temporary storage where we can fetch the data from the database we can perform some operation and save so that the data is altered as per the execution of query.

query to delete emp details where eno is 101

~~select * from~~

with tememp as

(select * from the emp where eno = '101')
delete from tememp

when we execute the above query first the inner query will get executed and the result of the inner query will be stored in CTE i.e. here 101 details will be stored in tempemp and then delete command will delete 101 from tempemp.

* write a query to delete duplicate records from the table?

~~with tempemp (ename, empcount)~~
~~as (select ename, Row~~
~~row-number()~~⁽¹⁾~~over (partition by ename)~~⁽⁴⁾
~~order by ename) as 'empcount' from~~
~~emp2)~~⁽²⁾
~~delete from tempemp where empcount >~~⁽³⁾
~~1~~⁽⁶⁾

In the above query. the inner query:

INDEXES

are used for fast accessing of data.

→ By using indexes we can improve the performance of a table.

① Clustered Index

② Non clustered index

③ Clustered Index - when ever we apply

primary key constraint on a column in a table then automatically clustered index will be applied on primary key column and then clustered index will arrange the records in ascending order.

→ create table emp(eno int primary key, ename varchar)

In the above query we apply pk on eno col then automatically clustered index will be applied column.

+Table

temp

+ columns

+ eno(int, PKay, Not null).

+ indexes

+ PK- eno(Clustered)

Note: we can apply only one clustered index on a single table.

→ Syntax for applying clustered Index:

Create clustered Index Index-eno on e4(eno)

⑧ Non clustered Index: whenever we apply unique key constraint on a column in a table then automatically non clustered index will be applied on ^{unique} column.

→ we can apply more than one non clustered index on a single table.

Syntax: Create ^{non} clustered Index Index-ename
uq_il on es(ename)

* Identity columns- If we want to apply auto generate-ID in sql server we have to use Identity column.

Syn: Create table 'eno' tablename(column ^{datatype} name
datatype Identity(Starting no, increment by),
columnname datatype)

Ex: Create table emp(eno int Identity
(101, 1), ename datatype)

→ we can't insert the number for the Identity column.

10/12/14

TSQL : Transact

It is a programming language in SQL Server
of TSQL program is

```
[declare]  
Begin  
end
```

declare blocks: This block is used to declare variable
begin block: This block is used to set the value
for the variables and we can write some logic within the begin block

end block: This block is used to end the program.

syntax to declare the variable:

@variablename datatype.

ex: @eno int

 @ename varchar(20)

syntax to set the value for the variables:

we can set the values in two ways

① by using set command

② select command

```
declare  
@a int  
@b int  
begin  
set @a = 10  
set @b = 20  
print @a @b  
end.
```

write a program to declare two numbers

```
declare  
@a int, @b int  
begin  
set @a = 10  
set @b = 20  
set @a = @a + @b  
set @b = @a - @b
```

set, @a=@a - @b
 print @a
 print @b
 end

print /'the value of '
 print 'the value of a is' cast(@a as varchar(20))
 print 'the value of b is' cast(@b as varchar(20))

- * write a program to declare student no, student name, marks1, marks2, marks3 calculate Total marks, percentage of marks and display them

declare

@sno int	begin
@sname varchar(20)	set @sno = 101
@m1 int	set @sname = 'Anil'
@m2 int	set @m1 = 100
@m3 int	set @m2 = 100
@total int	set @m3 = 100
@percentage float	set @total = @m1 + @m2 + @m3
	set @percentage = @total / 3.0

print

print /'the & %'

print 'sno is :'+cast(@sno as varchar(20))
print 'sname is :'+cast(@sname as varchar(20))
print 'total is :'+cast(@total as varchar(20))
print 'percentage is :'+cast(@percentage as varchar(20))

- * Bids Bookname, Quantity , price

declare

@bid int	@total bill float
@bname varchar(20)	
@quantity int	
@price float	

begin

@bid set @bid = 101
set @bname = 'Anil'
set @quantity = 3
set @price = 100

set @total_bill = @quantity * @price.

print 'Bill is \$'

print 'Total bill is' + cast(@total_bill as
varchar(20))
end.

* write a program to declare two numbers
and check which greater.

declare

@a int

@b int

begin

set @a = 10

set @b = 20

if (@a > @b)

print '@cast(@a as varchar(20)) is big'

else

print cast(@b as varchar(20)) + ' is big'

end.

* Create a ^{table} named student with columns
sno, sname, m1, m2, m3, total & percentage
write tsql program to calculate total marks
and percentage of marks and update the
total and percentage to student table.

+ create table

declare

@sno int

@sname varchar(20)

@m1 int

@m2 int

@m3 int

@total int

@percentage float

begin

set @sno = 101

set @sname = (select sname from student where
eno = @eno)

set @m1 = (select marks1 from student where
eno = @eno)

set @m2 = (select marks2 from student
where eno = @eno)

set @m3 = (select marks3 from student
where eno = @eno)

set @total = @m1 + @m2 + @m3

set @percentage = @total / 3.0

update student set total = @total,

percentage = @percentage where sno = @sno

print 'Record is updated successfully.'

end.

* create a table with name employee with
columns eno, ename, bsalay, da, hra, tsal.
calculate da, hra, Tsal, and update da, hra,
Tsal.

declare

@eno int @ename varchar(20).

@da float money

@hra float money

@tsal float money

@bsal float money

begin

set @eno = 101

set @ename = select ename from employee
where eno = @eno

set @bsal = select bsal from employee

where bsal eno = @eno

set @da = 0.2 * @bsal

set @hra = 0.4 * @bsal

set @tsal = @bsal + @da + @hra

QUESTION

Stored procedure :- is set of precompile SQL statement which will get executed when we call it.

→ stored procedure is a saved query whenever we pass SQL query three types of operations will be done

① Syntax checking

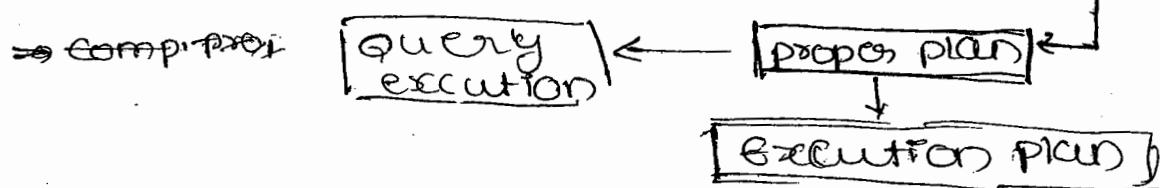
② Proper plan selected (any indexes are applied)

③ Query execution will be done. (on a table or not)

These three operations will be done everytime when we pass the pass the query.

Whenever we create a procedure then the syntax checking will be done and the proper ^{plan} will be selected and the proper plan will be stored in stored procedure and an execution plan will be stored in execution plan. whenever we execute the procedure then the query execution will be done from exception.

→ Create a procedure → [Save] → [Compile]



→ Stored procedure are of 2 types

① System defined

② User defined.

System stored procedure :- The stored procedure that are available when we install SQL Server Management Studio are called as system defined SP.

Always system defined procedures starts with [SP-]

→ The stored procedure that we create depending on the user requirement are called user defined stored procedure.

we must not declare user defined sp with
SP-:

Syntax:- create procedure ProcedureName(~~para~~
~~metas~~)

as begin
----- SQL statements

end
stored procedure will gets

syntax to call the procedure

① exec procedurename.

* when we execute the procedure the "procedure" will get executed from execution plan.

Note: at the time of execution of procedure we have to pass the values.

- the no of values that we pass must match with no of ~~parameters~~,
~~values~~

→ The order & type of ~~parameters~~ that we pass must match with ~~type of~~ order and type of parameters.

* write a program to perform addition of two numbers.

create procedure proc-add (@a int, @b int)

as begin

declare @c int

@c id set @c = @a + @b

print 'sum is' + cast @c as varchar(20)

end.

exec proc-add 10, 20

* write a program to declare the fname and lname and print fullname.

create procedure proc-fullname(@fname
varchar(20), @lname varchar(20))

as begin

declare @fullname varchar(50)

set @fullname = @fname + @lname

print 'full name is' + @fullname

end (calling)- proc-fullname 'sathya', 'Tech'.

* Create a procedure to insert the records in emp table.

```
create procedure proc-createemp (@eno int,  
@ename varchar(20),  
@salary money)
```

as begin

```
insert into emp values (@eno, @ename, @salary)  
end.
```

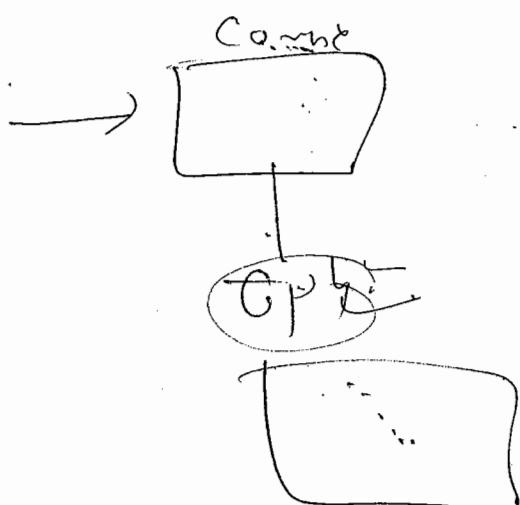
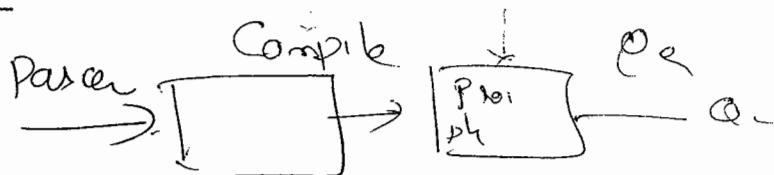
```
exec proc-createemp 101, 'sagar', 23000
```

* Create a procedure to delete the emp details based on eno?

```
Create Procedure Proc-deleteemp (@eno int)
```

as begin

```
delete from emp where @eno = @eno  
end.
```



SP is lot of
Pre Compiled SQL

which

P | T

foreach (Var i n in Gdt)

~~Select a row and value
into~~

* create procedure proc-updateemp(@eno int, @ename
varchar(20), @salary money)

as begin

update employee set ename = @ename,
salary = @salary where eno = @eno
end

calling :- [proc-updateemp 108, 'anil Kumar', 4000]

* create procedure proc-displayemp

as begin

select * from employee
end.

[proc-displayemp]

* create procedure proc-displayemp(@eno int)

as begin

select * from employee where eno = @eno
end.

[proc-displayemp]

* create procedure to display dnames based
on dno.

@ create procedure proc-dname(@dno int)

as begin

select * from dept where dno = @dno,

end

exec proc-dname 10]

* create procedure to display emp details
along with dno and dname from both
emp and dept tables.

@ create procedure proc-displayemp

as begin

select * e.* , d.dno, d.dname from emp
e inner join dept d on e.eno = d.eno
end

[proc-displayemp]

Q) Create a procedure to display emp details whose names starts with a?

→ Create procedure proc-disp

as begin

select * from emp where ename like 'a%'
end.

15/12/14

Q) Create procedure to display city names based on statename.

Create procedure proc-display (@sname)

as begin

select c.cname from state s inner join
city c on s.sid = c.sid where s.sname = @
sname

end.

[Proc-disp 'AP']

occurances
(20))

* Can we call one stored procedure in another stored procedure.

Create procedure proc1

as begin

select * from emp

end

→ Create procedure proc2

as begin

select * ~~from~~ ^{exec} proc1

end.

→ [proc2]

* select * from dept

end

exec proc2

S P
①
②
a O
O K
C V
a S
e d
e i
S C
H I
O S
e d
()
ce H
P
① Cu
ii
c

- Date: 14/10/2023
- * Stored Procedure will accept 2 types of parameters
 - ① Input Parameters
 - ② Output parameters : It is used to return a value from procedure.
 - Output parameters must be declared with `output` keyword.
 - * Create procedure `proc-disemp (@eno int, @ename varchar(20), output)`
- ```

as begin
 set @ename = (select ename from employee
 where eno = @eno)
end
declare @x varchar(20)
exec proc-disemp 102, @x output
select salary from employee where ename = @x

```
- \* Create a proc to display the no of male working in Company.

```

create procedure Proc-dispmale
 (@gender varchar(20), @empcount int output)
as begin
 set @empcount = (select count(*) from employee
 where gender = @gender)
end
declare @no.of.emps int
exec proc-get.emp by gender 'male',
@noofemps output print @noofemps

```

- \* Way to display .where
- How to Handle Exceptions in stored procedures

- ① what is a Exception ?

Exceptions will occur because of invalid input that was given by the user.

⑥ How to handle Runtime Errors in SQL by using Exception Handling Mechanism

16/01/2023

TC

Foc

⑥ Q1

Q2

dc

1.

2.

3.

4.

5.

6.

7.

8.

9.

10.

11.

12.

13.

14.

15.

16.

17.

18.

19.

20.

21.

22.

23.

Syntax: Begin Try

(10) ① ② ③

- - - - 1

- - - - 2

- - - - 3

End Try

Begin Catch

select error-message()

if(10 < 20  
it 3  
End Try

End Catch.

Try Commit  
Catch Rollback

Note: Here the statement will get executed even if exception causes i.e. the statement that we have written outside the catch block will get executed even if exception occurs.

Create Procedure Proc-div (@Int, @Bnt)

as begin

begin try

declare @c int

set @c = @a / @b

print @c

End Try

Begin Catch

Print ERROR-MESSAGE()

- Print 'Denominator must not be zero'

End Catch

Print 'R will execute even if Exception occurs'

END

End proc-div \$,0

## TCL Commands:-

Transaction Control language Commands.

Q) what is a Transaction?

- Ans:- Transaction is a piece of work that has been done. Different types of transaction commands
1. commit :-  $ctrl + s$  (Save changes permanently)
  2. Rollback :-  $ctrl + z$  (undo the changes)
  3. Savepoint:- Save the position of the transaction when ever we are performing any transactions then the operations will affect to buffer database but not for permanent database.  
- If we want to save the changes permanent database then we have to use commit

```

① select * from employee emp
 | delete from employee eno ename salary
 | where eno = 104 101 anil 20,000
 | rollback 102 sunil 23000
→ Syntax for transaction:-
```

begin transaction transaction name  
----- sql statements.

exit begin transaction t1

delete from employee where eno = 101

select \* from employee.

Observation:- when we execute the above query then 101 records will be deleted from buffer database

buffer data base  
emp

| eno | ename   | salary |
|-----|---------|--------|
| 102 | ensunil | 23000  |
| 103 | ajay    | 24000  |

permanent database  
emp

| eno | ename | salary |
|-----|-------|--------|
| 101 | anil  | 20000  |
| 102 | sunil | 30000  |
| 103 | ajay  | 24000  |

we

end.

- ① create a trigger to display a message saying that record is inserted successfully.

Q8

empty. X

Create trigger name on emp  
after insert  
as begin  
print 'Record is inserted'  
end  
insert into emp

create

- ② create trigger to display a message saying that record inserted successfully?

→ create table emp(eno int, ename varchar(20))

→ create trigger trig-emp on emp

after insert

as begin

print 'Record is inserted successfully'

end.

insert into emp values(102,'sunil')

when we execute above query then the record will be inserted in table and the trigger will be invoked immediately after performing inserting operation and display a message saying that record is inserted successfully.

- ③ create a trigger to display employee details after inserting the record in emp table?

create trigger trig-display on emp

after insert

as begin

select \* from emp

end.

insert into emp(103,'ajay').

- \* Create a trigger to change the lowercase to uppercase and update in table.  
 whenever we are working with triggers internally to tables (magic tables) will be created named inserted and deleted these are magic tables.
  - Inserted table will be executed immediately after performing inserting operation.
  - Deleted table will be executed afimmediately after performing deleting operation.
  - These two magic tables can be used only with in the triggers.
  - create trigger trig-uppers on emp  
 after insert  
 as begin  
 declare @eno int, @ename varchar(20),  
           @desig varchar(20)  
 set @eno = (select eno from inserted)  
 set @ename = (select ename from inserted)  
 set @desig = (select desig from inserted)  
 update emp set ename = upper(@ename),  
           desig = upper(@desig)  
 where eno = @eno  
 select \* from emp  
 end.
- \* when we execute the above query the record will be inserted into emp table and the trigger will be invoked and inserted table will be executed, and record is inserted in to inserted ~~table~~ <sup>isql</sup>.
- we have to declared three variable and ~~be~~ select the values from the inserted table and store in variable and then we have to convert the lower case characters

into upper case and update the values to emp table

emp

| eno (PK) | ename | desig |
|----------|-------|-------|
|          |       |       |

empaddress

| eno(FK) | address |
|---------|---------|
|         |         |

\* Requirement: when we want to insert the record in emp then automatically insert the record in empaddress table

\* Create trigger trig - Create emp on emp  
as be after insert  
as begin

@declare @eno int

set @eno = (select eno from inserted)

insert into empaddress(eno) values(@eno)

select \* from emp

select \* from empaddress

end.

→ insert into values(103, 'Anil', 'dev').

\* Cursors:- Cursor is a temporary sql memory area which is used to fetch the data from the table and return multiple rows.

Steps to work

① Declare the cursor

② Open the cursor

③ Fetch the data from the cursor

④ Close the cursor

⑤ Deallocate the cursor.

\* Create a cursor to display employee name and salary

declare empcus cursor for

select ename, salary from employee

declare @ename varchar(20), salary money

open cursor

fetch next from empcus into

@ename, @salary while

@@FETCH\_STATUS = 0

begin

Print 'Salary of ' + @ename + ' is ' +  
cast(@salary as varchar(20))

fetch next from empcus into @ename,

@salary

end

close empcus

deallocate empcus.

## UNIVERSITY ADMINISTRATION

The main aim of the project is to develop a user-defined webapplication to maintain the university Administration system.

### Modules:

- ① Department: This module is used to maintain the details like Departmentname, budget, startdate, location.
- ② Course: Every Department consists of some courses and each course will have credits.
- ③ Instructor: Instructor is the person who will teach the course.  
one Instructor can teach multiple courses  
one course can taught by multiple instructor  
Instructor will have fname, lname, hiredate, address
- ④ Student: Student is the person who will enroll the course in the university  
one student can enroll multiple course.  
and multiple students can enroll one course

Persons ( pid , int )

1  
2  
3  
4

departments ( deptn of ) Varchar ( 10 )

dep-1

dep 2

dep 3

4

Course cd-1 Varchar ( 10 )

g) Primary key ( col1, col2 )

17  
10/10/14

Views: view is an imaginary table or virtual table.

- ② view is used to hide the confidential information that is available in the table
- ③ view is used to provide abstraction.

view are of two types

① simple view

② complex view:

① simple view :- The view that was created on single table is called as simple view.

e.g:- Create view viewname.  
as

select query

Create a view to display emp details working in IT department.

execute view itdept

as

select \* from emp where dname = 'IT'.

\* With check option:- is used to restrict the inserting operation on view.

In the above view IT manager can't insert any other row in table.

→ Create view itdept

as

select \* from emp where dname = 'IT'

with check option.

\* Insert into values(222, 'sugar', 'Rohit', 30, 'hr')

When we execute above query then it will display error message saying that the target view is specified with check option.

④ Complex view:- The view that was created on single table multiple table.

**emp**

| eno | ename | salary |
|-----|-------|--------|
| 101 | Anil  | 20000  |
| 102 | Bunil | 23000  |

**dept**

| d_no | dname | eno |
|------|-------|-----|
| 10   | IT    | 101 |
| 20   | HR    | 102 |

- \* Create a view to display employee details along with department name.

Create view itdept

as

Select e.\* , d.dname from emp e inner join dept d  
on e.eno = d.eno where dname = 'IT'.

- \* select \* from itdept

- ⑥ Way to insert the record on complex views.

insert into itdept values(104, 'sagar', 23000, 'IT').

(Error)

Note: When we want insert values in the instead of complex view then we have to go for a concept called & instead of triggers.

⇒ Create trigger trg\_itdept on itdept

instead of insert

as begin

declare @eno int, @ename varchar(20),

@salary money, @dept varchar(20)

set @eno = (select eno from inserted)

set @ename = (select ename from inserted)

set @salary = (select salary from inserted)

set @dname = (select dname from inserted)

insert into emp values (@eno, @ename, @salary)

@dname,

insert into dept(dname, eno) values (@dname, @eno)

end

insert into itdept values (105, 'Exam', 23000, 'IT')

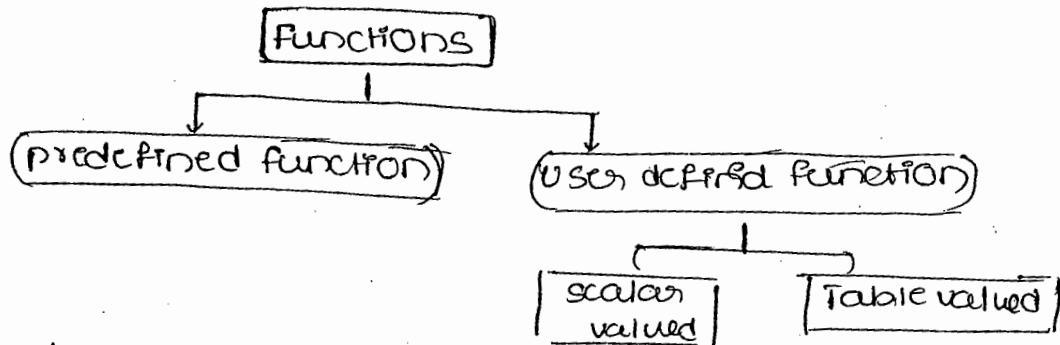
When we execute the above insert query instead of insert query instead of triggers will execute.

i.e first the record is inserted in inserted table

7/12/14

## functions

It is used to perform some operation and return some value of any datatype.



user defined function: - The function that was created created by the programmer depending on the user requirement is called user-defined function.

① scalar valued function: - The function which will take input from the user and return only one value of any datatype is called as scalar value function.

Syntax for creating scalar valued function.

① create function functionname (parameters)

returns datatype.

as begin

declaration

execution

return variable  
expression

end

\* Function will get executed when we call it.

Syntax for calling function:

[select dbo.functionname (values)]

dbo = database object

\* Create a function to accept two values from user and return sum.

create function fadd (@a int, @b int)

returns int

as begin

declare @c

set @c = @a + @b

return @c

end.

calling the function: select dbo.fadd(10,20)

② create a function to display employee name based on eno.

create function Fename(@eno int varchar)

returns str varchar(20)

as begin

declare @ename varchar(20)

set @ename = select ename from employee  
where eno = @eno.

③ return @ename

end.

\* select dbo.Fename(10)

\* Table valued function- The function which will

take input from user and return multiple rows

from the table. Is called as table valued function.

The return type of table value function is Table.

Syntax: Create function functionname(parameters)

returns table

as begin

return(select query)

Syntax to call table valued function:- select \* from

Select \* / collist from dbo.functionname(values)

e.g:

create a function to display emp details based on eno?

create function Function1(@eno int)

returns table

as begin

return (select \* from dbo.#employee where  
eno = @eno)

select \* from dbo.Function1(103).

## scalar valued function(SVF) | table valued function(TVF)

- |                                                                                                                                                                                                                                                                                                                               |                                                                                                                                                                                                                                                                                                                                                    |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> <li>① will process on single row and return only one value.</li> <li>② The return type of SVF is datatype.</li> <li>③ SVF will return a value, variable / expression.</li> <li>④ SVF will have as begin end</li> <li>⑤ SVF syntax to call SVF is select dbo.function name (x,y)</li> </ul> | <ul style="list-style-type: none"> <li>① TVF will process on multiple rows and return multiple rows from table.</li> <li>② The return type of TVF is table.</li> <li>③ TVF will return a table <u>select query</u></li> <li>④ will not have begin end.</li> <li>⑤ syn to call TVF is select * / collist from dbo.functionname (values).</li> </ul> |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

### stored procedure

- ① SP is set of precompiled SQL statement which will gets executed when we call it
- ② SP will compile only one time at the <sup>time of</sup> creating the procedure
- \* ③ SP will have execution plan
- \* ④ SP will support exception handling
- \* ⑤ SP will support DML commands
- \* ⑥ support select command
- \* ⑦ SP will support T-SQL commands like commit, rollback and save point

### function

- ① function will compile every time and it is used to return the value.
- ② function will compile every time.
- \* ③ function will not have execution plan
- \* ④ function doesn't support exception handling
- \* ⑤ function doesn't support DML commands
- \* ⑥ support select command
- \* ⑦ function doesn't support T-SQL Commands

- ⑥ SP may or may not have I/O parameters.
- ⑦ Function must have at least one I/O parameters.
- ⑧ SP can be executed by using exec command.
- ⑨ Function must execute with select command.
- ⑩ SP will accept both I/O and O/I parameters.
- ⑪ Function will accept only I/O parameters.
- ⑫ SP may contain 0 or more parameters.
- ⑬ Function must contain at least 1 parameter.
- \* ⑭ We can call one SP in another SP.
- ⑮ We can call one function in another function.
- \* ⑯ We can call function in SP.
- ⑰ We cannot call SP in function.

### How to take the backup of SQL Server Database :-

~~18/11/14~~

Triggers:- Triggers is a special type of stored procedure which will gets invoked immediately after performing DML operation like Insert, Delete, Update.

Different types of triggers are:-

① DDL triggers

② DML triggers

③ Instead of triggers

DDL triggers:- These triggers are used to create /alter/drop the table.

Syntax:-

create trigger triggername on tablename  
after insert  
as begin