SQL

SQL - Structured query language

SQL is a domain specific language and data base tool which is used to create and access database to support sw application.

Database → collection of data
Systematic

DBMS → collection of programs which enables its users to access database manipulate data, help in representation of data.

Ex: FB, Instagram, telephone directory, Electricity service provider.

First → 1960 → Charles Bachmen's → First DBMS is IDS
↓
Integrated data store

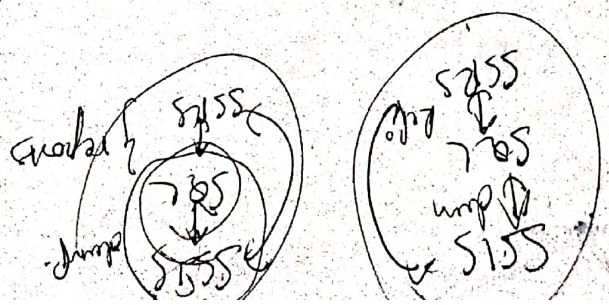
MSBI - Microsoft business Intelligence.
↓
Which provides best solutions for Business intelligence & data mining queries.

MSBI →
- SSIS → SQL Server Integration Services
- SSAS → Analysis
- SSRS → Reporting tool.

Normalisation

3

⇒ we should not store any tables in tempd in system databases because if SQL gets restarts



Scanned by CamScanner

System database:-

→ All system related objects which gets stored in master database.

→ ~~we shouldn't store any tables~~

Model data base:-

→ Default model values for all the objects in SQL Server will comes from model data base.

MSDB = Microsoft database → objects related to SQL Server jobagent
↓
SQL Server Agent

& few other things will get stored in MSDB.
↓
related to back up & restore information

tempdB = To store all temperory table.

→ It is faster becoz it won't write Log file in database
↓
Becoz it is recreating

If exists (select Top1 from deleted)
begin
set @action = 'Record is deleted'
end.

Scanned by CamScanner

→ Install it in mixed mode. (windows authentication)
↓
extra step for security.

→ filestream → storage of files.

→ Tabular mode.

History:

2005 - SQL Server 2005
(SSIS & SSQS)

2016 - SSMS is seperated.

2017 - SSRS is seperated.

Data types:-

1. Numeric data types:-

→ TINYINT → to store every record it requires 1Byte (0 to 255)

→ SMALL INT → 2Bytes → 2^{-15} to $2^{15}-1$

→ INT → 4bytes → 2^{-31} to $2^{31}-1$

→ BIG INT → 8bytes → 2^{63} to $2^{63}-1$.

→ In Tiny int we cannot store negative values

→ we cannot store decimal points. (using float (or) real we can store decimal points).

FLOAT:- To store decimal points.

Real → Ansii SQL.

Float → Microsoft SQL

Decimal:-

DECIMAL(P,S) P → Precision Ex: dec(10,2) ⇒ (234567
Ansii S → Scaling

Microsoft → Numeric(10,2)

Scanned by CamScanner

Character data types:- There are

1. CHAR → Fixed length → 1 byte for each character (max (n) = 8000)
2. VARCHAR → Variable length → 1 byte for each char + 2 bytes for over head take. (Max(n) = 8000)
3. NCHAR → Fixed length → 2 bytes for each char (max(n) = 4000)
3. NVARCHAR → variable length → 2 bytes for each char + 2 bytes of head → column name storage. (max(n) = 4000) over ~~take~~

CHAR:

In CHAR it is fixed length i.e in char (10)

① we cannot go beyond 10 characters

② If we insert 'A' also i.e one character it will take 10 bytes.

(I.Q) Diff b/w char, nchar, varchar, N varchar.

CHAR	VARCHAR	NCHAR	NVARCHAR
→ It is fixed length	→ variable length	→ fixed length	→ variable length.
→ Non unicode	→ Non unicode	→ unicode	→ unicode.
→ max leng = 8000	→ max leng = 8000	→ maxlen = 4000	→ maxleng = 4000

⇒ These take twice as much storage space, so its better to use only if you need unicode.

→ SQL Server page can store upto 8192 i.e 8KB = 8 × 1024.

SQL Server by default has page header it stores 96 bytes

= 8192 − 96

= 8096.

③ BIT datatype To store '0's & '1'

Binary → To store binary values (fixed length)

VARBinary → To store varbinary (variable length)

only DATE → DATE

TIME → TIME.

Scanned by CamScanner

→ DATE TIME - less nano seconds → (DD hr : mm : sec . nn)

→ DATE TIME 2 - more nano seconds

IMAGE → To store images

TEXT, NTEXT

XML → Extra markable language

Money → to store monitory values.

Note:-
In a select statement we can put 'N' no. of columns for displaying but while storing into the table we can keep upto 1024 columns

⇒ varchar(max) ⇒ Lobfiles.

5/12:

SQL Server:-

MS SQL Server

Syntax for creating a database:-

Create database ~~Name~~ Database_name

(words which are in blue color are Keywords)

Ex:-
Create database Bank

Note:-
⇒ we cannot use Keywords as database names, column names, table name

⇒ whichever the Keywords looks like blue (or) black color those are called "Keywords"

How to execute a Query (or) commands:-

⇒ Select the query ⇒ press F5 (or) control + E.

(or)

press Execute button

Syntax to use a database:-

use database name Ex: use Bank

Scanned by CamScanner

Note:- When we create a database it will create two files in the backend

1. Master data file (.mdf)
2. Log data file (.ldf)

=) In a Server we can create around 65K databases.

Commands:- These are used to communicate (32,767) with the database to perform specific task.

1) DDL (Data definition Language):- used to define the database structure

1. Create 2. Alter 3. Drop 4. Truncate 5. Rename.

2) DML (Data Manipulation Language)

1. Insert 2. Delete 3. update 4. Merge, ~~Select~~.

3) DQL (Data Query Language) => used to retrieve the data from database

1. Select 2. print

Select → o/p in result window
print → o/p in message window.

4) ~~DCL~~ TCL (~~Data~~ Transaction control Language) => used to control access to data stored in a database

1. commit 2. Save (or) Save point 3. Roll back.

5) ~~TCL~~ DCL (~~Transaction~~ data control language) => used to manage the changes made to the data.

1. Access 2. Denied 3. Revoke

Syntax for creating a table:-

```
Create table tablename
(
col1 datatype
col2 data type
col3 data type)
```


Scanned by CamScanner

Ex:-

Create table Tbl_Employee details.

(

Eid int,

Ename varchar (255),

EDOB Date,

E Sal Float,

E Add varchar (500),

Emailid varchar (100),

EDes varchar(100),

EGender char(1)

EMobil Bigint,

is Active Bit)

Syntax to get all tables in database:-

1. sys.tables

① select * from sys.tables

② Select * from sys.objects where type='U'

③ Select * from sys.sysobjects where type='U'

Note:-

Table ~~columns~~ names (or) column names should not start with integers, Special characters and also we should not give space in the Keywords.

⇒ If we want to make above criteria as true we need to put the object name in a Square brackets.

⇒ whenever we retrieve the data from that object it is mandatory to put the Square brackets, if we use integers (or) Special characters

Note:-

⇒ we can create 1024 columns in a table.

⇒ In terms of SQL, field names (or) column names are called Structure (or) metadata

Scanned by CamScanner

Syntax to retrieve the data:-

Select * col1, col2, col3 from tablename

(or)

Select * from tablename

(or)

Select top (10) * from tablename

(or)

Select top (5) percent * from tablename

(or)

Select top (5) Salary from Tablename.

Syntax to insert the data into a table:-

Case(i):- Insert into tablename (col1, col2, col3) values ('val1, val2, val3)

(or)

Insert into tablename values (val1, val2, val3)

Syntax to insert the multiple ~~values~~ rows using single insert statement:-

Insert into tablename values (val1, val2, val3), (val4, val5, val6), (val7, val8, val9)

Ex:

Insert Tbl- Employee details

Val1 ('2', 'Madhu', '1994-05-20', 20000.00, 'NRT', 'xyz@gmail.com', 'GW', 'M', 1234565, 1)

b

Scanned by CamScanner

Note:

1. When we are inserting the data into a table the no. of passing values should be equal to no. of columns which we have mentioned while creating the table.

2. The flow of passing the values should be same as ~~your~~ our column flow.

3. If you are not passing any value for a particular column the system will defaultly consider as Null value.

* Null is defined as nothing, it won't compare with any value also null itself

Null x anything = Null.

Syntax for updating a new value:-

update tablename set Email id = 'madhu@gmail.com' where Eid = 1.

update tablename set Email id = 'madhu@gmail.com', isactive = 1 where Eid = 1.

Note:-

→ using single update statement we cannot update multiple rows but we can update multiple columns corresponding to that row.

→ Before updating any value make sure that the expected o/p is coming (or) not in the result stat by writing a query, if yes then replace with update, otherwise rectify your query.

Syntax for renaming a database:- Alter database olddatabasename modify name = new databasenew.

Alter Database = old database name.
Modify Name = New database name.

Ex: Alter database = Flipcart
Modify name = Flipkart

Scanned by CamScanner

Syntax for renaming a tablenames:-

Sp- Rename 'oldtableName', 'New ~~old~~ table name'

Ex: Sp- Rename 'Epl-Employee details', 'Epl - Employee'

Sp - System defined store procedure → The bunch of statements which will be in the sw Ex: Rename.

Syn: Sp- Rename 'tablename.oldcolumnname', 'New column', 'column'.

Ex: Sp- Rename 'Ecde tbl-Employee. Edes', 'EDesignation', 'column'.

Note:

In a table multiple column should not contain Same name.

Syntax for Adding new column for an existing table:-

Alter table table-name

Add Newcolumnname datatype.

Ex: Alter table table-Employee

Add Emid int, E Homenumber Varchar(100)

Ex: Eid EName ESal EDOB - - - - Emid E Home number

1. John 10000 - - - - - - - - NULL ~~[illegible]~~ ~~[illegible]~~ NULL

Syntax for changing the datatype for an existing column:-

Alter table tablename

Alter column column name Newdatatype.

Ex: Alter table ~~[illegible]~~ tabEmployee

Alter column ESal Money.

Scanned by CamScanner

→ Note:-

1. By using a single alter statement we can add 'N' number of columns at a time.

2. By using a single alter statement we can change only one datatype in a column at a time.

Syntax for dropping an existing column from the table:-

Alter table table_name

Drop column column_name.

Ex: Alter table table_Employee

Drop column Emid, EHomenumber.

Eid	EName	- - - -	Emid,	EHN
1	John	- - -	Null	Null

Syntax for dropping an existing table:-

Drop Table TableName1, TableName2, TableName3

Drop Table T1, T2, T3 —

Note:-

1) when you drop a table the complete structure with data will be deleted permanently and you cannot retrieve back the deleted data.

2) cannot shuffle when we add columns in an existing table, columns will be added at the end of the table by default. we cannot add in the middle (or) in the first of the table.

→ we cannot shuffle the columns also in a table.

Scanned by CamScanner

IMP I.Q.

Delete	Truncate
1. Syntax: DELETE FROM TABLE_NAME (or) DELETE TABLENAME.	1. Syntax: TRUNCATE TABLE TABLE_NAME.
2. When we do delete (or) truncate only data will be get deleted and Structures remains same.	
3. Incase of delete we can do condition based deletion. Ex: Delete from Table name where id ≤ 20.	3. We cannot do condition based deletion incase of truncate.
4. When we delete the data from a table with (or) without where condition, it will always lock each & every ~~dep~~ row & ~~delete~~ perform the deletion operation.	4. When we truncate the data from a table it will lock the whole table & truncate in a single shot.
5. Compared to truncate deletion will be much slower.	5. Compared to truncate deletion will be bit ~~slower~~ faster.

Scanned by CamScanner

According to 2008 R_2 version we can rollback the delted data using transactions.

⇒ According to 2008 R_2 version we cannot roll back the delted data (No Log file)

7. According to 2012 version we can rollback both the deleted & truncated data.

8. When we del the data in an identity table the next seed value will be depending on last inserted value.

8. When we truncate the data in an identity table. the seed value will be reset to '1'.

9. DML

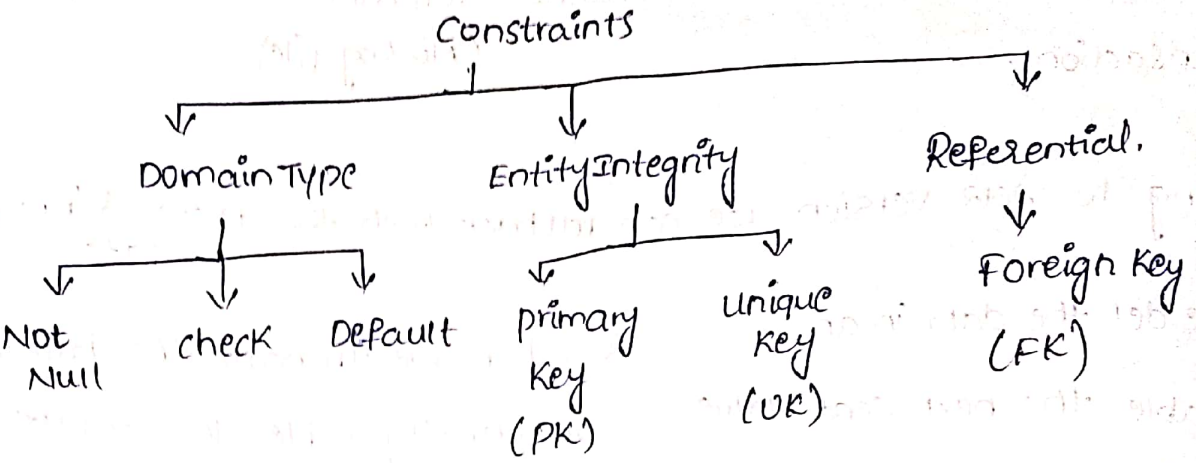
9. DDL.

Syntax for dropping the database:—

Drop database databasename

Scanned by CamScanner

Constraints:-



⇒ constraints are nothing but set of conditions (or) properties which we are applying before & after creating the table.

Not NULL CONSTRAINT:- Ensures that column cannot have a NULL value.

case (i):- while creating a new table

```
create table tbl_product
(
 pid int Not NULL,
 pname varchar(100) Not NULL,
 pcost float,
 pQty int,
 is available Bit Not NULL
)
```

Ex: Insert table_product values (1, 'A', 10, 3) ✓

case(ii) :- Insert For an existing table

case(i): without data

If there is no data in the table directly we can apply

Scanned by CamScanner

the below syntax to apply NotNULL constraint for an existing column

Alter table table-name
Alter column column-name datatype Not NULL

Ex Alter table table-product
Alter column pid int Not NULL.

Case (ii) : ① Existing ~~data~~ table with data:-

⇒ when we have a data in a table check whether that particular column is having NULL's (or) not if yes

Alter table table-name
Alter column column-name datatype Not NULL

⇒ If yes

⇒ we cannot directly apply the Not NULL constraint on top of the column

Note: If we want to apply Not NULL on top of this first we have delete null values then apply the syntax.

syntax: Alter table table-name.
After column columnname Not NULL

⇒ we can apply NotNULL constraint more than one.

Default constraint:-

⇒ If we want to put a constant value (or) static value ~~for a particular~~ then we will go ahead with default constraint

Note: ⇒ Default can apply on top of any column and any datatype

case(1):- while creating New table:-

Scanned by CamScanner

```
Create table Tbl-product
(
  Pid int Not NULL,
  PName Vchar(100), Not NULL,
  Pcost float Default 100,
  Pqty int,
  is Available Bit Default 1,
)
```

Note:

When there are 'N' no. of columns in a table and for one particular column having default constraint then we need to pass n-1 values.

→ Incase of defaut constraint it is mandatory to mention required columnname while inserting the data.

→ If we want to insert an external value for the default column we have to mention all the columnnames and pass the correspond values so that it will choose an external value instead of default value. ⇒ whenever we mention an external value i.e it will take default value.

Insert into Table product name (pid, pname, pqty) values (2, B, 10)

Case 2:- Existing table without data:-

Syntax:- Alter table Tbl-name
Add constraint DF-product-pcost
default 200 for pcost

Scanned by CamScanner

Case 3:- Existing ~~data~~ table with data:-

Same as above syntax but default value won't be applicable for existing data and it will be applicable for newly inserting records.

⇒ when you apply a default constraint, insertion will be different i.e syntax will be same but we need to add some column value names (or) remove column value names.

System defined databases:-

1. Master data base
2. Model data base
3. MSDB
4. Temp DB.
5. Resource DB ⇒ Hidden database

Schemas:-

→ When we create an object (table, view, function--) defaulty system will assign one schema called dbo(data base object)

Note:

If we want to maintain two (or) more tables with the same tables name then we will ~~specify diff~~ create different schemas append to ~~the tabl~~ it while creating the table.

→ Incase of giving the access to a particular object admin will directly provide the access on top of particular schema so that developer can able to access to all the objects which are belongs to that particular schema.

Scanned by CamScanner

Syntax for creating a schema:-

create schema schemaname.

→ In a database we can create 'N' no. of schemas

→ When we are performing any operation on top of any object which are having different schemas, make sure that, that scheme name should be there before the object name.

→ If we want to drop the exisiting schema first drop all the dependency tables then drop your schemas ⇒ we cannot rename the schema.

Operators:-

⇒ =, =! (or) < >

⇒ <

⇒ >

⇒ <=

⇒ >=

⇒ in

⇒ not in

⇒ like

⇒ Between

⇒ AND

⇒ OR.

⇒ To hide the result screen ⇒ CTR+R

⇒ To ~~see complete information~~ know complete information of particular object. ⇒ ALT+f1

17

Scanned by CamScanner

8/12

Check constraint:-

⇒ By using check constraint, we can insert the data ~~by~~ into a table on a condition based.

⇒ If conditions satisfies, it will allow the record otherwise it will throw an error.

Case(i):- while creating a new table

```
Create table voterlist
(
vid int NotNULL,
vName VARCHAR(10) NotNULL,
vAGE INT Default 18,
vAge limit int Not NULL check( vAge limit >= 18 and vAgelimit <= 70),
is active Bit Default 1
)
```

Ex: INSERT INTO voterlist (vid, vname, vagelimit) values (~~1~~ vishy, 65)

Case(ii):- For an existing table :-

i) without data:-

```
Alter table ~~tabrename~~ voterlist
Add constraint CK_voterlist_vAgelimit
check (vAgelimit >= 18 and vAgelimit <= 70)
```

with data:-

→ Before applying a check constraint in an existing table which is

Scanned by CamScanner

having some data, we need to first check whether all the rows are satisfied with condition (or) not if yes we can use the alter syntax to apply check constraint if no either we have to change the condition based on minimum (or) maximum value (or) we can delete the data which are beyond (or) less than that limit then we can apply the check constraint

⇒ Note:- check constraint can be apply only on top of integers, float, money, date, columns.

⇒ We can apply more than one constraint on top of a single column.

⇒

Entity Integrity:-

primary key:-

To provide an identification for a table we will go with primary key

⇒ properties of primary key-

1. primary key will sort the data in physical order (ascending order)

2. primary key doesn't allow any duplicate records and null records as well.

3. when we create a primary key clustered index will get create automatically.

4. For a table we can able to provide only one primary key

Scanned by CamScanner

Because of shuffling the data.

Ex)

Id	Id1
2	9
3	7
9	2
6	1
5	4
4	5

(PK) ① (PK) ②

2	1
3	2
4	4
5	5
6	7
9	9

top(3) →

2	1
3	2
4	4

→ not matchable

If we select top 3 from two primary keys then we will get wrong result.

⇒ we can apply primary key on top of more than one column which will be known as composite primary key.

Note:
In case of composite primary key duplicates & null records will be depending on all the columns which are included in primary key.

Case(i) while creating a newtable:-

syntax:

```
create table tbl-Employee
(
Eid int primary key,
EName VARCHAR(10),
ESal Float
)
```

composite primary key

```
create table tbl-Employee
(
Eid int,
Mid int,
EName VARCHAR(10),
ESal Float,
primaryKey(Eid, Emid)
```


Scanned by CamScanner

For an existing table:-

without data:-

Alter table tbl_Employee

Add constraint PK_Employee_Eid

primary key (Eid)

For an existing table without data:-

Alter table table_Employee

Alter column Eid int Not Null

⇒ when we create a primary key for an existing table which is having no data, make sure that the column should be not nullable column

⇒ Alter table tbl_Employee

Add constraint PK_Employee_Eid

primarykey (Eid).

⇒ For an existing table with data:-

→ while creating a primary key for an existing table which is having some data make sure that key column data should not contain any null values & duplicate also make that column as not nullable.

⇒ Syntax will be as above.

⇒ ~~the~~ syntax for dropping a constraint:-

Alter table table-name

Drop constraint constraintname

} By using a single alter statement we can drop ~~both~~ multiple constraints

Scanned by CamScanner

Unique Key:-

→ To make two (or) more columns as distinct we will go with unique key

Properties of unique key:-

1. It allows only one null value and doesn't allow any duplicate records.

2. For a table we can keep 'N' no. of unique keys.

3. If you put unique & Not Null It will almost act as primary key which is also known as light weight primary key.

Case(i):

While creating a new table:-

```
Create table tbl_Employee
{
Eid int primary key,
Emid int unique Not Null,
Esal Float
).
```

⇒ For an existing table without data:-

```
Alter table tbl_Employee
Add constraint uk_Employee_Mid
unique (Mid)
```

With data conditions:

→ only one null should be there and no duplicates should be there.

→ If there are no duplicates and only one null we use same above syntax.

Scanned by CamScanner

Foreign Key :-

⇒ To maintain the valid data b/w tables we will create a primary foreign key relationships.

⇒ without primary key we cannot able to create foreign key

⇒ If a column having unique & not null then we ~~can~~ can take this column as reference, we can create relationship b/w foreign key & unique + not null column.

⇒ using a single primary key we can create 'N' no. of foreign keys.

properties of foreign key :-

1. It will allow both null & duplicate values.

2. Note :-

1. when there is a primary foreign key relation b/w two tables and while we are trying to insert the data into foreign key table first it will check whether that ID is present in primary key table (or) not. If yes it will allow if no it will throw an error saying that violation of primary key.

⇒ If we want to delete any records in primary key table first we have to delete dependency records in all foreign key ~~records~~ tables then only we can delete from primary key.

⇒ The other case to delete the record from primary key using cascade operation.

⇒ when there is a primary foreign key relationships we cannot directly drop the primary ~~for~~ key

Scanned by CamScanner

⇒ when there is primary foreign key relations we cannot apply truncate both primary & foreign key relationships.

⇒ only we have to use delete command to delete the data.

⇒ we can give relationship b/w different tables and within the table as well.

While creating new tables using primary & foreign:-

Case(1): using primary key

```
create table Accountdetails
(
AId int primaryKey,
AName varchar(100),
AAdd varchar(100)
)
```

using Foreign Key

```
create table transaction details
(
AId int foreignKey References
        Accountdetails (Aid),
Id int primary key,
AName varchar(100),
)
```

For an existing table:-

without data:-

```
Alter table transactiondetails
ADD constraint FK-transaction-Aid
                details
Foreign Key (Aid) References Accountdetails (Aid)
```

with data:-

⇒ whenever we want to apply primarykey foreignkey relationships for an existing table which is having some data then make sure that primary key column should not have null values & duplicate records also the data which is present in foreign key table should also present in primary key table as well.

Scanned by CamScanner

on cascade update / on cascade delete:-

→ If we want to delete/update the records from both primary key & foreign key tables at a time we can go with cascade option.

→ we should apply always cascade operation of foreign key, we should not apply on primary key.

Ex:

```
Create table Accountdetails_cascade.
(
AID   INT primary key
ANAME  Varchar(10),
ADOB   DATE,
AAdd  varchar(10),
Agender Char(1)
)

Create table Transactiondetails_cascade
(
AID INT ~~primary~~ Foreign key References Accountdetails_cascade
                (AID) on delete cascade | ON update cascade
TID INT primary key,
Tamount Float,
T Date Date
);
```

DELETE CASCADE:-

```
Delete from Account details_cascade where AID=5;
```

UPDATE CASCADE:-

```
UPDATE Account details_cascade set aid=5 where aid=1
```


Scanned by CamScanner

Example:

Accountdetails

Aid	Aname	Aaddress	A.mob
1.	A	NRT	9182
2.	B	GTR	8196
3.	C	VJW	9681

P1

Transaction details

AId	TId	Tamount	Tdate
1.	101	1000	12/1/2000
2.	102	2000	4/5/2009
3.	103	3000	5/20/2018

F1 P2

creditcard details

AId	CId	Ccategory	climit
1	C1	Platinum	1,00,000
3	C2	gold	2,00,000
3	C3	Silver	50,000

F2 P3

Loandetails

AID	LID	Lcate	Lamount	Lilenure
1.	L1	Persona	10,00,000	5
2.	L2	Home	2,00,000	10
3.	L3	Car	3,00,000	15

F4 P5

Loan Transactions

Lid	Lpaid	Lamount	Lpaid date
L1	3000	7/1/2000	
L1	5000	8/1/2000	
L2	7000	9/5/2000	

F5

creditcard transaction details

cid	CTid	CTamount	CTdat
C1	1001	50	3/2/12
C1	1002	100	4/3/14
C3	1003	150	5/16/17

F3 P4

Note: we cannot change datatypes of an existing column primary & foreignkey.

→ the only possibility is first drop both the constraints then append the datatype then change the datype and then add.

Scanned by CamScanner

Set operators:- To combine two (or) more different result set into a single result set we will go with setoperators.

1. union
2. union ALL
3. Except (or) Minus
4. Intersect.

Prerequicities:-

1. If we want to apply any of the set operators to any tables, first we have to check all the table structure will be same (or) not if yes we can able to apply the setoperators.

2. the corresponding column datatype should also match.

union ALL:- ~~unionAll allows two tables having different structure~~

→ It will combine two (or) more tables data into a single result set and it won't eliminate any duplicates & null records.

→ It will not sort the data also.

EX:

A

Id1	Name
1	A
2	B

B

Id2	Name
3	C
4	D

C

Id3	Name
5	E
6	F

Select * from C
UNION ALL
Select * from B
union all
Select * from A

Id	Name
5	E
6	F
3	C
4	D
1	A
2	B

Scanned by CamScanner

Note: The final result set column names will be depending on first select statement.

union operation: –

→ union will combine two (or) more result sets and it also removes the duplicate records as well. It allows NULL.

Note:

→ Duplicates are depending on no. of columns which are are taking in the select class.

Ex:

```
Select * from A
union
Select * from B
union
select * from C
```

A

Id1	Name1
1	A
2	B

B

Id2	Name
1	A
3	C

C

Id3	Name
1	A
4	D
NULL	NULL

A

	ID	Name
①	1	A
	1	A
	1	A
②	2	B
③	3	C
④	4	D
	NULL	NULL

Imp Note:

Compared to union, union ALL will be must faster because there is no sorting and removation of duplicates, no grouping, apply the distinct operation to ignore the duplicates in union, so it will take some time to perform all operations in backend.


```
Select ID1 from A
union
Select ID2 from B
union ~~ALL~~
Select ID3 from C
```

1
2
3
4
NULL

A
B
C
D
NULL

```
Select * from A
union
Select * from B
union ALL
Select * from C
```

1	A
2	B
3	C
1	A
4	D

Intersect:

→ It will give common records from two (or) more tables.

① A B

1	A
2	B
3	C

② A B

1	A
1	A
2	B

```
⇒ Select * from A
Intersect
Select * from B
```

1	A
2	B

③ A B

3	C

```
Select * A
Intersect
Select * B
Intersect
Select * C
```

⇒

ID	Name

⇒ In the back end it will collect all the duplicates and give result only one record.

⇒ It also eliminates the duplicates.

Scanned by CamScanner

Except:- Except will give the records which are present in first table but not same records present in second table.

Id	Name
1	A
2	B
3	C
4	D

Id	Name
1	A
5	f
6	g
7	h

Select * from A

~~Intersect~~ Except ⇒

Select * from B

2	B
3	C
4	D

⇒ It will sort the data and removes duplicates.

Examples- Query:-

Eid	EName	Esal	EDep
1	a	100	IT
2	b	250	IT
3	c	350	DBA
4	d	450	Admin
5	e	550	Admin
6	f	650	Admin

IT → 5%

DBA → 8%

Admin → 7%

(Esal * 0.05) + Esal

Select (Esal * 0.05) + Esal Hike, * from Employee where EDep = 'IT'

union ALL

select (Esal * 0.08) + Esal Hike * from Employee where EDep = 'DBA'.

union ALL

Select (Esal * 0.07) + Esal Hike * from Employee where EDep = 'Admin'.

Scanned by CamScanner

Imp

How to find whether two (or) more tables are in sync (or) not.

Employee

Eid	Ename	Esal	EDes
1	a	250	IT
2	b	350	IT
3	c	450	DBA.
4.	d	550	Admin
5.	e	650	Admin.

Employee-1

Eid	Ename	Esal	EDep
1	a	250	IT
2	b	350	IT
3	c	450	DBA
4	d	550	Admin
5	e	650	Admin.

```
Select * from Employee
Except
Select * from Employee-1
```

} Eid Ename Esal EDep.

Note:- If the above statement will gives empty result set then we can consider both the tables are in sync

→ If if it giving atleast single row then both the tables are not in sync.

Identity property:-

→ It will generate automatic generation number using seed & increment values.

→ Seed value can be changed at any point of time but increment value is fixed.

Note:-

If there are 'N' no. of columns in an identity table we need to pass 'n-1' values

Insert into Tbl-Employee values (A, 100, ABC)
(B, 200, DEF);

Syntax:-

```
create table Tbl-Employee
(
EmpNO INT Identity (1, 1)
Empname varchar(100),
ESal float,
EAdd varchar(100) )
```

increment value

seed value, Base value, starting value, original value

Scanned by CamScanner

→ If I want to insert an external value instead of '3' in the above table we do have two options.

1. we can Reseed the value by using dbcc command
↓
database check constraint.

Syntax:

DBCC check ident(TableName, Reseed, value)

Ex: DBCC check ident (Tbl-Emp, Reseed, 99)

⇒ If we want to insert the next number as '100' instead of '3' we need to reseed the value to 99. because the seed value will be always depending on increment value.

→ After executing dbcc command we need to follow 'n-1' formula to insert the values.

→ INSERT into tbl-Employee values (c, 250, Hyd), (d, 350, hyd)

Select * from tbl-Employee

Eid	EName	ESal	EAdd
1	A	100	Hyd
2	B	200	Hyd
100	C	300	Hyd
101	D	400	Hyd

Option 2:

→ By enabling identity insert property we can enter an external value & the last inserted value will be treated as seed value.

Syntax: Set Identity-Insert tbl-Employee ON

Note:- when we are enabling identity insert property in an identity table make sure all column names should be present while inserting the data.

Scanned by CamScanner

Ex: Insert into Tbl-Emp (Eid, EName, Esal, EADD) values (101, E, 350, Bgl), (1000, G, 450, Bgl).

Eid	EName	Esal	EADD
1	A	100	Hyd
2	B	200	Hyd
100	C	300	Hyd
101	D	400	Hyd
101	E	500	Bgl
1000	F	600	Blr

Set Identity_Insert Tbl_Employee off ⇒ After disabling the identity insert property we need to follow (n-1) formula to insert the values.

Insert

Note: For a table we can provide only one identity property.

→ From 2012 onwards microsoft has introduced one more concept called sequence which can be given more than one for a table

→ If there is a single column in a table that too identity then we need to insert the following statement to insert the values

Select Ident_current 'table name'

```
Create table T1
(
id int identity(1,1)
)

Select * from T1
```

Insert T1 Default values

Go 100.

Global variables

To find last inserted value in an identity table we can use global variable called @@Identity.

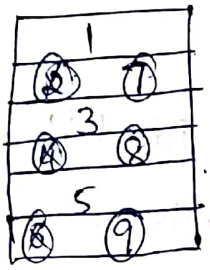
Id
1
2
3
⋮
100

We can also find using Select scope_Identity() (or). This comes under system defined functions

Scanned by CamScanner

1
2
3
4
5
6

Delete * T_1 where Eid= 2, 4, 6. then.



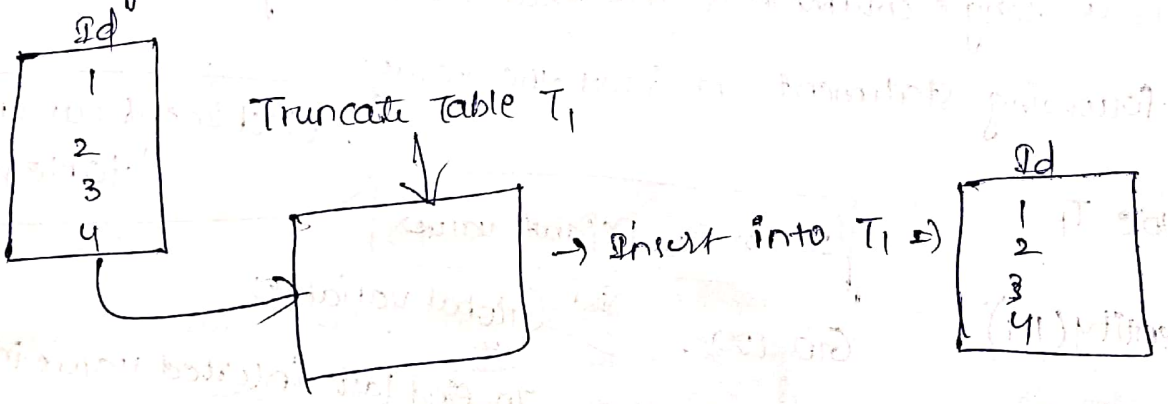
⇒ when we delete the data in an identity table the only data will be get deleted from the table and stores in log-file, also the memory allocation remains same.

⇒ when you insert a new value again first it will insert in the place of deleted rows then it will go to bottom.

Insert into T_1 ⇒ default values will be inserted

→ when we delete whole table & inserts a new value, the new value will be depending on last inserted row

⇒ when we truncate an identity value it will reseed the value to '1'. which means it is changing the definition of a table thats the reason it is coming under DDL command.



Note

⇒ we cannot update identity column.

⇒ we cannot drop identity property once we define in the table.

⇒ we cannot apply an identity property for an existing column.

Scanned by CamScanner

⇒ we cannot apply Identity property only on top of integer columns.

⇒ we can apply Identity property for a newly adding column in an existing table.

⇒ we can apply both identity & primary key in a single column.

→ we can use only one identity column per table.

→ we can keep negative value in identity and can keep negatives values as seed in increment value but it is not recommened in realtime.

→ In a single session we can enable Identity-insert property ~~only~~ at a time for a single table only. If we want to enable for other table as well in the same session first we have to disable for the first table then enable for second one.

→ we can enable Identity-insert property for two tables at a time in two different sessions.

How to take schema & ~~inser~~ data from a table:-

→ Right click on database

→ Go to tasks

→ click on Generate scripts.

→ Choose objects

→ check in Select specific database objects

→ click on toggle symbols of tables

→ Select required tables.

→ click on next

→ click on advanced tab

→ under the option of types of data to script → select either data only Schema and data (or) schema only options.

After table tablename

Add columnname datatype Identity property.

Scanned by CamScanner

→ If we want drop and create scripts we can choose under the option called script, drop & create.

→ click on OK.

→ choose the path to save the file.

I.Q:

Select * from T_1
Select * from T_2
Select * from T_3

} Here it will throw an error.

Let us assume in the above statement, B table is not exists in database and when we are trying to execute all the three commands at a time, it will execute first table, when it comes to second table it will throw a error beco2 table doesn't exists, also it will stop the execution process so It won't execute third statement

Select * from A
go
Select * from B
go
select * from C.

} A & C will be executed
B will throw an error.

⇒ when we ~~insert~~ keep 'go' keyword in b/w statements and if we are executing all the statements at a time. It will execute each and every statement even though above statement is success (or) failure.

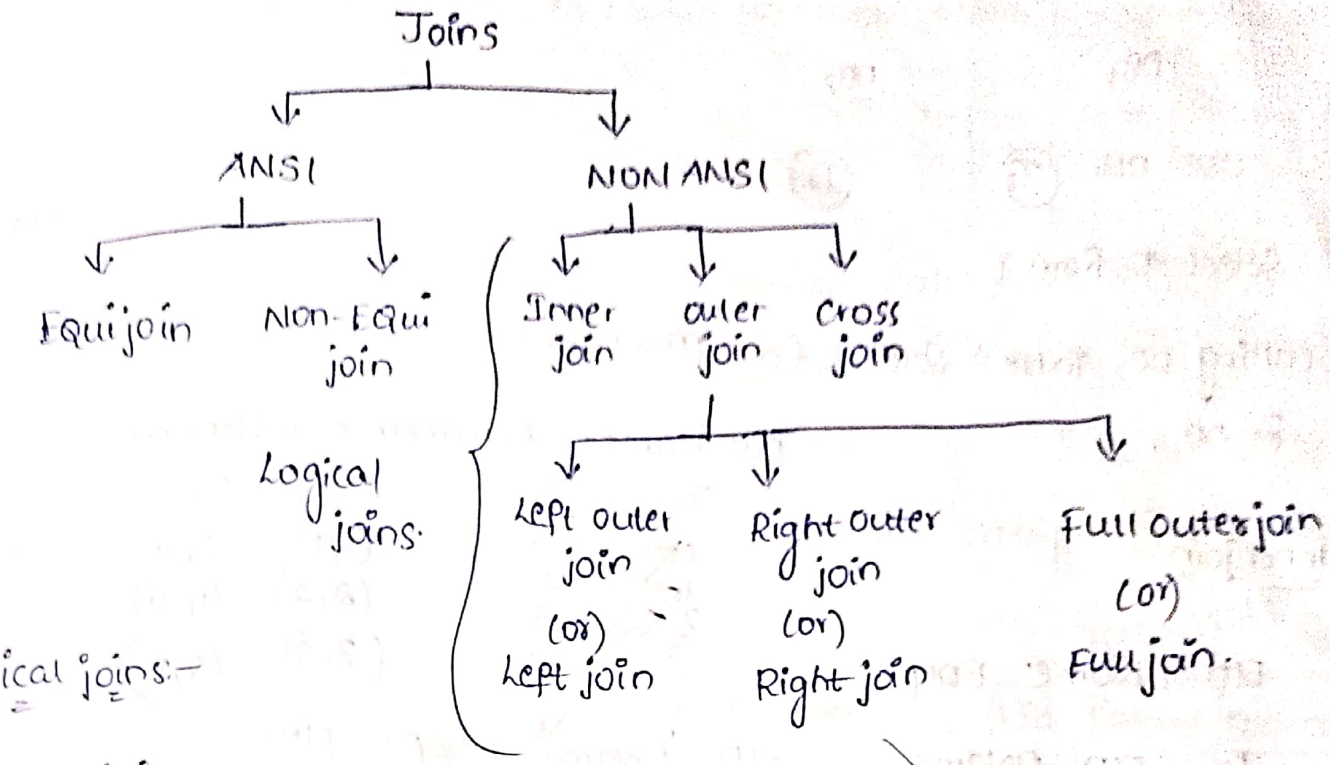
Note: Before set operators we cannot able to use orderby clause ~~of~~ beco2 of execution flow.

Select * from T_1 orderby id Asc
union ALL
Select * from T_2

Select * from T_1
union ALL
Select * from T_2
orderby id Asc

Scanned by CamScanner

Joins:-



Physical joins:-

1. Merge join
2. Hash join
3. Nested loop join.

join:-

To retrieve the data from two (or) more tables we will go to joins.

pre-requisities:-

1. To perform joins b/w two (or) more tables we require atleast one key column (It might be integer, varchar date)
2. Keycolumn datatype should match b/w both tables.
3. joints can perform both on relational & non relation tables.
4. Structure of both tables ~~might~~ may (or) ~~might~~ may not be same.
5. joins can be performed on two tables (or) more

Scanned by CamScanner

→ How to select the data from different databases.

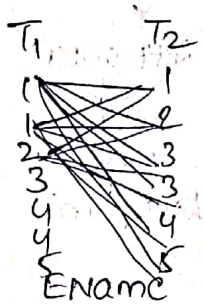
DB_1 DB_2

Use DB_1 T_1 T_2

Select * from T_1

Calling DB_2 Table ⇒ Selec * from DB_2.DBO.T_2
in DB_1.

Datename. Schemaname. Tablename.



(1,1) (3,3)
(1,1) (4,4)
(2,2) (4,4)
(3,3) (5,5)

Inner join:-

~~EID EName EDep.~~

	DID	DName	EID	EName	EDep.	DID
~~1.~~	~~ECE~~	~~1~~	~~Madhu~~	~~ECE~~	~~1~~	
~~2.~~	~~EEE.~~	~~2.~~	~~vishali~~	~~CSE~~	~~2.~~	
~~3.~~	~~Mech.~~	~~3.~~	~~Pavan~~	~~ECE~~	~~1.~~	

when we perform join b/w two tables T_1, T_2 each & every record in T_1 table will map with all the records in T_2 table and if it matches, it will pull the data and those records are called as matching records

Note:
If one record is ~~matching~~ matching multiple times in T_2 table, it will bring also bring duplicate records as well.

1. 1
1. 1
2. 2

count Total records = 3

Syntax: ~~select from~~ T_1.id In Select T_1.id.

Scanned by CamScanner

Inner join:-

→ when we perform inner join b/w two (or) more tables it will always give matching records only.

Syntax:-

Select * from Table A as A Innerjoin TableB as B ON.

A.column1 = B.column1.

Ex:

write a Query to get EmpID, Ename, and their corresponding department names.

department

DID	DName
1	ECE
2	EEE
3.	IT
4.	CSE
5	Mech.

(PK)

Employee

DID	EID	EName
1.	1	Karthik
1.	2	Rithvik
2.	3	Naithik
2.	4	Sonik.
4.	5	Bonic
5.	6	Tonic

Resultset

EId	EName	EDepar.
1.	Karthik	ECE.
2.	Rithvik	EEE
4.	Bonic	CSE
5.	Tonic	Mech.

Select E.Eid, E.name, D.DName ~~from Dept~~

from

Department as D inner join Employee E. ON

D. Did = E.Did

(or)

Select E.Eid, Ename, D.DNam from employee as E

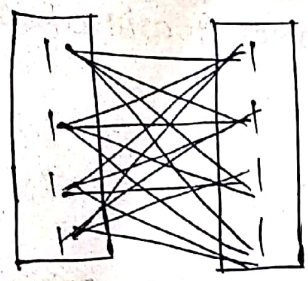
join

Dept D on E.Did = D.Did.

These two statements will give same result set in case of innerjoin

Scanned by CamScanner

I.Q:-



⇒ 4x4
= 16 records.

(1, 1)	(1,1)	(1,1)	(1,1)
(1, 1)	(1,1)	(1,1)	(1,1)
(1, 1)	(1,1)	(1,1)	(1,1)
(1, 1)	(1,1)	(1,1)	(1,1)

Left outerjoin:-

when we perform left outer join b/w two (or) more tables will get matching records from both the tables and unmatched records from left table. For those unmatched records it will assign null records for all colm in right table.

Formula:

left outer join = inner join + unmatched records from left table.

Syntax

Select E.Eid, E.Ename, D.DName from

Dept D left join Employee e on D.Did = E.Did.

Eid	EName	DName
1.	Karthik	ECE
2.	Rithvik	ECE

Scanned by CamScanner

write a query to get department names which are not having atleast a single employee. where E.Eid is NULL (or) E.Ename is NULL (or) E.Did is NULL.

Select E.Eid, E.Ename, D.Dname from Dept D left join Employee E on D.Did = E.did. where Eid is NULL

Result

Eid	Ename	Dname
NULL	NULL	N/w.

write a query to get all department names which are having atleast one employee without using inner join. where E.Eid is not null (or) E.Ename is not null (or) E.Did is not NULL.

Select E.Eid, E.Ename, D.Dname from Dept D left join Employee E on D.Did = E.DID where E.Eid is not NULL

Right outer join:-

If we perform right outer join b/w two or more tables it will give matching records from both the tables and unmatched records from right table and for those unmatched records it will assign NULLs for all the columns in left table.

Right outer join = Inner joint unmatched records from right table

Select E.Eid, E.Ename, D.Dname from ~~Dept D~~ Employee E. Right join Dept D on D.Did = E.Did.

Eid	Ename	Did	Did	DName
1	a	101	101	IT
2	b	101	102	DBA
3	c	102	103	support
4	d	102	104	N/w.
5	e	102		
6	f	103		

Scanned by CamScanner

Joins:

Examples:

Write a Query to get Employee Id & Employee name their correspond department names

Department

Did	Dname
101	IT
102	DBA
103	Support
104	Network

Employee -

Eid	Ename	Did
1	a	101
2	b	101
3	c	102
4	d	102
5	e	102
6	f	103

1. 101 101 IT
2. 101 101 IT
3. 102 102 DBA
4. 102 102
5. 102 102
6. 103 103

Select ~~E~~ · Select E · Eid, E · Ename, D · Dname from Department as D join Employee E ON D · Did = E · Did

Result ·

EId	EName	DName
1	A	IT
2	B	IT
3	C	DBA
4	D	DBA
5	E	DBA
6	F	support

Left join

Right

Left

Scanned by CamScanner

14/12/18

Full outer join.

" Full outer join = Innerjoin + unmatched records from left + unmatched records from right"

⇒ when we perform full outer join b/w two (or) more tables will get matched records from the tables and unmatched records from both left & right tables.

⇒ For those unmatched records, it will assign null values correspondingly

Ex:

	ID	ID1	Result ID	Result ID1	
M.R ←	1	1 → M.R	1	1	Inner join.
	2	3 → M.R	3	3	
M.R ←	3	5 → M.R	5	5	
	4	9 } U.M	2	NULL	Left join.
M.R. ←	5	10. }	4	NULL	
			NULL	9	Right join.
			NULL	10	

Select * from A full join B ON A.ID = B.ID1

full join = Innerjoin − Left + Right

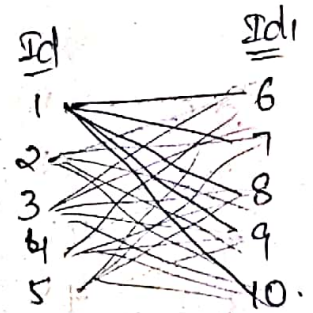
Cross join:-

Syntax:-
Select * from A cross join B

Select * from A, B.

It is a cartesian product which will give all possible combination of records.

Ex:



(1,6)	(2,6)	(3,6)	(4,6)	(5,6)
(1,7)	(2,7)	(3,7)	(4,7)	(5,7)
(1,8)	(2,8)	(3,8)	(4,8)	(5,8)
(1,9)	(2,9)	(3,9)	(4,9)	(5,9)
(1,10)	(2,10)	(3,10)	(4,10)	(5,10)

Select * from A,B. (5x5) = 25 records

Scanned by CamScanner

Note: There won't be any "ON" condition incase of cross join.

Equijoin:-

Equi join will give matching records from both the tables and it almost act like inner join only.

syntax:

Select * from A, B where A.id = B.id1

A	B
1	1
2	3
3	5
4	6
5	9

(1,1)
(3,3)
(5,5)

Result

1	1
3	3
5	5

Select * from A, B where A.id = B.id1.

Non Equijoin:-

It will give all possible combination of mismatched records b/w both the tables

syntax:

Select * from A, B where A.id <> B.id

(or)

A.id != B.id.

Id	Id1
1	1
2	1
3	2
4	4
5	5

Result

1 2	2 5	4 1	5 2	
1 4	3 1	4 1	5 4	
1 5	3 1	4 2		
2 1	3 2	4 5		
2 1	3 4	5 1		
2 4	3 5	5 1		

Scanned by CamScanner

⇒ joins for more than two tables:-

A	B	C
Id	Id_1	Id_2
1	1	1
1	1	2
2	3	3
5	5	4
3	7	5
	8	

⇒ Select * from A Innerjoin B ON A·Id = B·Id_1 Left join C ON B·id_1 = C·Id_2

Innerjoin Result:

1	1
1	1
1	1
1	1
5	5
3	3

Id_2
1
2
3
4
5

Leftjoin Result

Id	Id_1	Id_2
1	1	1
1	1	1
1	1	1
1	1	1
5	5	5
3	3	3

Right join Result

Id	Id_1	Id_2
1	1	1
1	1	1
1	1	1
1	1	1
3	3	3
5	5	5
NULL	NULL	2
NULL	NULL	4

⇒ Select * from A· Innerjoin B ON A Id = B·Id_1 Right join C ON B·Id_1 = C·Id_2.

⇒ Select * from A Leftjoin B ON A·Id = B·Id_1 full join ON B·Id = C·Id_2.

A	B
Id	Id_1
1	1
1	1
1	1
1	1
5	5
3	3
2	NULL

	B·Id = C·Id_2	
1	1	1
2	1	1
3	1	1
4	1	1
5	5	5
	3	3
	NULL	NULL
	NULL	NULL
	NULL	2
	NULL	4

A·Id	A·Id_1	C·Id_2	A·Id
1	1	1	1
1	1	1	1
1	1	1	
1	1	1	
5	5	5	5
3	3	3	
	NULL	NULL	2
	NULL	NULL	4

Scanned by CamScanner

AId	BId1	CId2
1	1	1
1	1	1
1	0	0
1	1	0
2		

a Select *. from A Full join B ON Full join ON. B.Id1 = C.Id2.

A Id	B.Id1	C.Id2
1	1	1
1	1	1
1	1	0
1	1	0
2	NULL	
NULL	0	

A Full join B

A.Id	B.Id1	C.Id2
1	1	1
1	1	1
1	1	1

Self joins:-

⇒ If we apply a join within the table then it will be known as Self join.

Ex:

Employee

Eid	Ename	Mid
1.	Karthik	NULL
2.	Rithvik	1
3.	Naithik	1
4.	Sonik	2
5.	Arthik	4

Manager

Eid	Ename	Mid
1.	Karthik	NULL
2.	Rithvik	1
3.	Naithik	1
4.	Sonik	2
5.	Arthik	4

Select E.Ename as EName, M.EName as Managename from Employee E left join Employee M ON E.Mid = M.Eid.

Query: write a query to get all the employees and their respective managers.

Q.2: write a query to get employee names who are having atleast one manager.

⇒ Select E.Ename as EName, M.EName as ManagerName from Employee E Innerjoin Employee M ON E.Mid = M.Eid.

where M.Eid is not null

(or)

M.Ename is not Null

(or)

M.Mid is not null.

Scanned by CamScanner

Write a Query to get list of employee names who are not having any manager

Select E.Ename, as employeename, M.Ename as Managername

from Employee E ON join E.mid = M.Eid where M.Eid is NULL

(or)

M.Ename is NULL

(or)

M.Mid is NULL

=>

Employee

Eid	EName	Mid
1	John	NULL
2	peter	1
3	Kevin	1
4	Kemp	2
5	Thomas	3

Manager

Eid	EName	Mid
1	John	NULL
2	peter	1
3	Kevin	1
4	Kemp	2
5	Thomas	3

Olp

EmpName	Manager Name
John	NULL
peter	Thomas
Kevin	Jhon
Kemp	peter
Thomas	Kevin.

Scanned by CamScanner

15/12/18

Id	Id1
NULL	NULL
NULL	NULL
NULL	NULL
NULL	NULL
0	NULL

I.J = 0

L.J = 5

R.J = 5

F.J = 5+5 = 10.

18/12

Temporary tables:-

→ To store the data temporarily we will go with temporary tables.

There are two classifications in it.

1. Local temporary table.
2. Global temporary table.

Imp Note:

→ when we insert the data in temporary tables it won't ~~write~~ Rigl write any log files so that performance wise temporary tables will be much faster.

⇒ when we restart the SQL services tempdb will be dropped and recreates ~~the table~~ again.

Scanned by CamScanner

: 12 —

1. while creating a table if you put one # keyword, (or) one pound before the table name then it will be known as Local temporary tables.

Syntax:-

```
create table #T1
(
CXID INT,
cxName varchar(10)
```

→ Insert into #T1 values(1,'Madhu').

→ Select * from #T1

2. Incase of Local temporary tables, we can create multiple tables with the same table name in different sessions. These tables are stored under tempdb & to differentiate these

Global:-

1. while creating a table if we keep '##' keyword ~~then~~ before the tablename then it will be known as Global temporary tables.

Syntax:-

S1
```
Create table ##T1
(
CXID INT,
CXName varchar(10)
)
```

S2 Insert into ##T1 values(1, 'Madhu').

S3 Select *. from ##T1

2. Incase of Global temporary tables, we can not create multiple tables with the same table name in different sessions. If we create a global temporary table in one session that will be accessible in all Remaining sessions.

Scanned by CamScanner

tables System will assign random numbers while creating the table.

Ex: #T1 ---- 001F
#T1 ---- 002F.

③ If we create a Local temporary table one Session, it will be restricted to that particular Session only. So we cannot ~~to~~ able to access in any of other sessions.

④ wherever, we created a Local temporary table that session is nothing but "parent session". once we close that session automatically that temporary ~~tempdb~~ tables which are created under that Session will get dropped.

③ Incase of Global temporary tables, it won't assign random numbers. So we can access in remaining Sessions also.

④ Incase of Global temporary tables until unless we close the parent session the table remains exists in the database. once we close the parent Session, ~~you~~ we cannot able to retrieve that tables in any of the child Sessions.

Scanned by CamScanner

Global:-

Session-1	Session-2	Session-3
Create table #T1 (CID INT) Insert into #T1 values()	Create table #T1 (CID INT) Insert into #T1 values	Create table #T1 (CID INT)

Global:

S1	S2	S3
Select * from ##T1	Create table ##T1 (CID INT)	Insert into ##T1 values (1).

parent session

child tables.

Scanned by CamScanner

Note:-

⇒ Always try to drop the temporary tables manually by writing a query becoz, if we keep that session in open state, consumed memory won't get released until unless we close that session.

⇒ we can apply all the DDL & DML operations on top of temporary tables.

⇒ we cannot apply primary & foreign key relationships for temporary tables

⇒ we can create Indexes on top of temporary tables.

⇒ we can apply Identity property on top of temporary tables.

⇒ we cannot use temporary tables inside a view & user defined function as well.

⇒ we can apply joins on temporary tables.

⇒ we can join both temporary with physical tables

⇒ we can use temporary tables inside a stored procedure.

⇒ we can apply use all constraints except referential integrity (but foreign key table will get create without reference) Table will be get create

Imp

⇒ How to dump the data from one table to another table

Syntax:-

Select * into newtable from oldtable } for one time dump.

Ex: Select * into #Employee from Employee ↓ we cannot add data multiple times

Note:-

when we are executing above query it will copy whole structure as well as data from employee to #Employee table. It won't carry forward any of the constraints from oldtable to newtable.

⇒ when we execute this above query multiple times it will throw an error stating "object already exists". So this is a one time activity

⇒ If we want to capture only the structure instead of data we need to use below syntax,

Scanned by CamScanner

Syntax:-

1. Select * into #T1 from T2 where 1 = 2

2. Select top (0) * into #T1 from T2

3. Select col, col2, col3 into #T2 from T1.

Syntax to insert the data into an existing table:-

```
create table #T1
(
CXID INT,
CXName varchar(10))
```

Insert into ~~#Employee~~ #T1 select * from T2

(or)

Insert into #T1 select CXID, CXName from T2

Note:-

we can dump the data from physical to physical, physical to temp, temp to temp, temp to physical.

Table variable:- A table variable is declared using the table datatype

→ This is also one of the temporary object the scope of table variable will be till that execution only.

→ Initially it will store the data in RAM once it is filled it will store in tempdb.

→ we cannot drop tablevariable manually

Syntax

```
Declare @tablename table
(
col1
col2
```


Scanned by CamScanner

Ex

```
Declare @T1 Table
(EID INT,
Ename varchar(10)
Insert into @T1 values (1, 'Madhu').
Select * from @T1
```

It should be in single execution.

⇒ we cannot apply select * into for table variable

⇒ To insert the data from physical to table variable use the following

Syntax

```
Declare @T1 Table
( EID INT,
  Ename varchar(10))
Insert into @T1
Select * from @T1
Select * from @T1
```

Single execution.

Single statement

Limitations:-

1. we cannot apply alter statement for table variable.

2. we can use tablevariable in case of userdefined functions that to in multistatement table valued function.

3. we cannot truncate the table variable but we can delete it.

4. we cannot create indexs manually by using syntax.

Scanned by CamScanner

Local variable Temporary tables	Table variable
1. Here we will create the table Syntax:- create table #tablename(T1) (EID INT) Insert into #T1 Values(1) Select * from #T1	1. Here we will declare the table. In syntax first we will declare tablename & table. Syntax: Declare @tablename(T1)table (EID INT) Insert into @T1 values(1) Select * from @T1 } single execution
2. Here we can execute the commands seperately.	2. Here we have to execute all commands in a single execution step. After execution, table will drops automatically but it shows resultset.
3. It stores in tempdb	3. It stores in RAM memory.
4. Here we have to drop the table manually by using drop command.	4. Here we cannot drop (or) truncate the table becoz after execution table will be dropped automatically
5. If we want to load large amount of data from physical to any temp (tempdb) we will go for temporary tables.	5. Generally RAM will allocate some amount of memory to table variable, if we insert more than that allocated memory then it will be stored in tempdb. So when retrieving the data, it retrieves from both tempdb & RAM

Scanned by CamScanner

6. It is for large amount of data and entire data will be get stored in tempdb

which takes more time. So table variable is useful for lesser amount of data

Ex:- we have 4GB RAM, it allocated 500MB for table variable, if we insert 600MB data, 500MB will be stored in RAM & 100MB will be stored in tempdb while executing, it retrieves the data from both tempdb & RAM, RAM again has to work to get the data from tempdb.

~~6. we can execute only one time~~

6. ~~We can execute n no. of times~~

Uservefined variables —

→ A variable can store a single scalar value (either int (or) char (or) date)

→ The scope of variable will be till that execution only.

Syntax:-

```
Declare @id int, @name varchar(20)
Set @id = 1
Set @Name = 'abc'
        (or)
Select @id, @Name
```

Single execution.

→ print @id, print @Name → It will print in messages.

```
Declare @date date = '01/01/2018'
```

Declaring a value to the variable from table:— 57

pid	pname	cost	pyr
1	A	10	2018
1	A	15	2019

```
Declare @cost float
Set @cost = select pcost from product table
where pid=1 and pyear=2018
```

Scanned by CamScanner

(or)

Set @cost = pcost from product where pid=1, and year=2018.

Select @ cost.

String functions: — A function which will retrieve a single scalar value in o/p i.e is called as "string function."

There are classified into:—

1. Left
2. Right
3. LTRIM
4. RTRIM
5. upper
6. Lower
7. Length
8. Replicate
9. Substring
10. charIndex
11. PhatIndex
12. Replace.
13. Stuff
14. Reverse

Left:—

Syntax: Select Left(expression, value)

↓ column name (or) string value (or) variable

Ex: Declare @city varchar(100)

Set @city = 'Banglore'.

Select Left(@city, ~~Banglore~~) = Bang.

Right

Syntax: Select Right(expression, value)

Ex: Declare @ city varchar(100)

Set @ city = 'Banglore'

Select Right (@city, Banglore) = Alore.

Scanned by CamScanner

Reverse:- Other way to retrieve the right characters.

Syntax: Select Reverse(expression)

Banglore

Select Reverse (Left(Reverse, (@ city), 5))

Banglore

Expression

o/p: erolgnaB.

Expression

glore erolg

Upper:-

Syntax: Select upper (Expression)

Lower:-

Syntax: Select Lower(Expression)

Ex: Declare @Name varchar(100)

Set @ Name = 'Karthik'

Select upper(@ name) = KARTHIK

Select Lower (@name) = karthik

Length:

Syntax: Length(expression)

Ex Select Length (@name) = 7 (Including space)

Error:- Length is not a recognised build in function name.

Select (upper(Left (@name, 1)) + Lower(Right (@name, Len ~~Length~~ (@name) -1))

K + arthik

Karthik

Scanned by CamScanner

LTRIM:- It removes ~~left~~ spaces from left side.

RTRIM: It removes spaces from right side

Syntax: Select LTRIM (Expression)

Select RTRIM (Expression)

Declare @string ~~varc~~ varchar(100)

Set @string = ' Banglore '

Select ~~length~~ (RTRIM(LTRIM (@string))) → Banglore

CharIndex:- a particular character position

(expression to ~~search~~ find, expression to search, start location)

Select charIndex (character, expression, Searching position)

optional parameter

Declare @Name varchar(10)

Set @Name = 'Apple'

malayal(a)m

Select charIndex (p, @name) → To find first 'p' position

To find Second 'p' position:-

Select char Index ('p', @Name, charIndex('p', @Name+1))

Searching position

Star after 1st 'p'

→ To findout second 'p' position find findout first 'p' position and add one value for it to change Searching position

Select charindex ('a', 'PaarthSamthaan') → first a position

Select charindex ('a', paarthSamthaan', charindex ('a', 'paarthSamthaan')

Scanned by CamScanner

Substring:-

Select substring (expression, starting position, Length)

Declare @ string varchar(10)

Set @ string = 'Soap_20_Soap'

Select substring (@string, charindex('_', @string)+1,

[charindex('_', @string, charindex('_', @string) +1)

- charindex ('_', @string) -1] 2nd '_' position

second '_'

1st '_' position.

Replace:- It will replace all similar characters with newstring

Length (oldstring) ≤ (or) ≥ Length(Newstring)

Length (Newstring) ≤ (or) ≥ Length (old string).

Syntax:

Replace (Expression, oldstring, newstring)

Declare @ string varchar(10)

Set @ string = 'Mblbybltbm'.

Set @ String = 'azle'

Select Replace (@string, 'b', 'a') (or) select Replace (@string, 'z', 'pp')

'Mandanna'

~~Replicate~~

To find first 'a' → charindex('a', 'Mandanna')

To find second 'a' → charindex('a', 'Mandanna', charindex('a', 'Mandanna'

To find third 'a' → charindex('a', 'Mandanna', charindex('a', 'Mandanna' charindex('a', 'Mandanna'+1) +1) 6(

Scanned by CamScanner

Stuff: It will replace a particular character position.

Select Stuff(expression, Special position, length, Newstring)

Ex AZlz

Select stuff(@string, 2, 1, pp)

⇒ Appl(z)

Select stuff(stuff(@string, 2, 1, pp), 5, 1, 'e')

Result ⇒ Apple

Replicate:—

Syntax: Replicate (Expression, value)

Select Replicate (Samthan, 3)

Result: SamthanSamthanSamthan.

IQ:

How to replicate the stars in pyramid format.

Scanned by CamScanner

Patindex:-

Select patindex(range, expression)

Select patindex('%[p-z]%', 'ABCD54321ABC') => Result → 0

=> patIndex will give first character (or) Integer position which will be in the given range.

Ex)

Select patindex('%[0-9]%', 'ABCD54321ABC') => Result = 5

Select patindex('%[A-Z][0-9]%', '8754ABCD321') => Result = 5

Select patindex('%[A-Z]%p^q^r^', '543PQRSABCD') Result → 4

↓ If we want to eliminate some characters.

Q1

Declare @string varchar(100)

@string = 'ABCD152345PQRS567' } get characters & Integers seperately using patIndex

=>

Set @position = patindex('[%a-z%]', @string)

while (@position) > 0

Begin

Set @string = Stuff(@string, @position, 1, '')

Set @position = patindex('[%a-z%]', @string)

print @string

end

Select @string.

Scanned by CamScanner

Query: T_1 From the below table get employee names & employee company names Seperately.

EmailId

T_1

Email Id

Madhu@Ey.com

Shashank@wipro.com

Karthik@Accenture.com

Nythik@wellsfargo.com

Rithvik@

~~Substring('Madhu@Ey.com', charindex('@', @EmailId'),~~

~~charindex('@', @EmailId +1)~~

~~Select substring('EmailId', charindex(Emailid', 1, charindex('@', Email)~~

from T_1

~~Select substring('EmailId', charindex('Email', charindex('@',~~

~~Select substring('EmailId', charindex('Em '@', EmailId'~~ —

charindex('@', 'EmailId'

~~Select substring('EmailId', charindex('@', EmailId' +1,~~

charindex('@', 'EmailId' +1) — charindex('.', 'EmailId') -1

Select substring('EmailId', charindex('@', 'EmailId')+1,

charindex('.', 'EmailId') - charindex('@', EmailId) -1]

from T_1

Scanned by CamScanner

Email Id

Declare @string varchar(20) = 'Email Id',

To get names before '@':

Select substring (EmailId, 1, charindex('@', EmailId) -1) from T1

To get names after @.

Select substring (EmailId, char index('@', EmailId)+1), [charindex('.', EmailID)

-charindex('@', EmailID) -1] from T1

Query: Name
a-b-o
a-
a-b-c
de-ef
b-
c-a-

To find - before '-'.

Select Substring (Name, charindex('-', name), charindex('-', Name + '-') -1]
from T1

Imp To get all A's in Malayam.

Select substring Len('@string') - Len(Replace(@string, A, '.'))

⇒ 8-4=4

Query: Set @string = 'A apple a day'

~~Stuff(Stuff(~~ Stuff('@string', 1, 4, 'AN') = An apple aday

Scanned by CamScanner

a-b-c-d-e → Find D position.

Select substring [@string, charindex('-', @string, charindex('-', @string, charindex('-', @string)+1)+1)+1, → Starting position.

[charindex('-', @string, charindex('-', @string, charindex('-', @string, charindex('-', @string)+1)+1)+1) - charposition('-', @string, charposition('-', @string, charposition('-', @string)+1)+1)) -1] ⇒ D position.

4th(-)

3rd(-)

22/12

Ranking functions:- These are used to write Querys. these are also called as window-function.

① Row_number()

② Rank()

③ Dense_Rank()

④ NTile(N)

Row-number():- To provide an automatic generation number temporarily for an existing table, we will go with rownumber.

Syntax:-

Select Row_number() over(order by column name ASC) As Rno, * from Table.

(or)

Select * Row_number() over(order by column name DSC) As Rno from Table

Note:-

Row number functions will generate serial numbers with the seed value of '1' and increment by '1'.

→ Eventhough there are duplicates (or) null records in an existing

Scanned by CamScanner

table, row-number will generate an increment value. so there won't be any duplicate numbers in row-number column.

Rank:-

Rank will assign ranks for an existing table with some gaps (or) with gaps.

Syntax:

Select Rank() over(orderby column name Desc) As Rank, * from tablename Employee

Ex.

Rank	SID	Sname	Smarks.
1	1	A	540
1	2	B	540
3	3	C	320
3	4	D	320
5	5	E	240

Syntax:

Select Rank() over(orderby Smarks desc) As Rank, * from Employee.

DensRank():-

DensRank() will assign ranks for an existing table without gaps.

Syntax:

Select DensRank() over(orderby column name Desc) As Rank, * from Table name

Ex:

SID	SName	Smarks
1	A	540
2	B	540
3	C	320
4	D	320
5	E	240.

Result:

Rank	Dens Rank	SID	Sname	Smarks.
1	1	1	A	540
1	1	2	B	540
3	2	3	C	320
3	2	4	D	320
5	3	5	E	240.

Syntax

Select Dens Rank() over(order by smarks desc), As Rank, * from Employee

Scanned by CamScanner

Select Row-number() over(order by sid Asc) As Rno, Rank() over(order by Marks Desc) As Rank, Dens_Rank() over(order by Marks Desc) As Dens_Rank * from Tablename.

Sid	Sname	Smarks
1	A	520
2	B	520
3	C	340
4	D	340
5	E	320

Rno	Rank	Dens	sid	sname	Smarks
1	1	1	1	A	520
2	1	1	2	B	520
3	3	2	3	C	340
4	3	2	4	D	340
5	5	3	5	E	320

NTile(n):- NTile will divide into no. of parts based on number which we have provided.

Select NTile(4) over(orderby id Asc) as Nfile, * from T1

Ex:

	NTile(4)		NTile(5)	
1		1		1
2	→ 3	1		1
3		1	→ 3	1
4		2		2
5	3	2	→ 2	2
6		2		3
7	2	3	→ 2	3
8		3		4
9	2	4	→ 2	4
10		4	→ 1	5

Note:- we cannot use where condition in Alias columns in a direct select statement beco2 of the execution of the query.

Select Row-number (orderby Sid Asc) As Rno, * From T1 where Rno = 3 ✗

It throws error in column rno

From Beco2 when we retrieve whole row there we cannot use where

where

Select

68

Scanned by CamScanner

Derived tables — In a from class if we write a select query then it will be called as derived table

```
Select A.empid from
(
Select Row_number() over(order by Eid ASC) AS Rno, * from Employee
) A where A.Rno = 3
```

Rno	Eid	Ename
1	1	A
2	2	B
3	3	C
4	4	D

⇒ Write a query to find 5th highest position.

EmpID	Empname	ESal
1	A	500
2	B	100
3	C	135
4	D	500
5	E	450
6	F	650
7	G	750
8	H	750
9	I	750
10	J	250

```
Select A.Esal from
(
select Dens_Rank() over(order by Esal DESC)
AS DR, *. from T1
) A where A.DR = 5
```

ID	name	Esal	DR
7	G	750	1
8	H	750	1
9	J	750	1
6	F	650	2
4	D	500	3
1	A	500	3
5	E	450	4
10	J	250	5
3	C	135	6
2	B	100	7

```
Select A.Ename from
(
Select Dens_Rank() over(order by
Esal DESC) AS DR, * from T1
) A where A.DR = 5;
```


Scanned by CamScanner

Name
Ram
Raj
Mahesh
Mohan
Anvesh
Avinash

Select * from T1 where Name like 'a%'

↓

The word starts with a.

%a → word ends with a.

'a%h' → Starts with a ends h.

'__a%' → ~~can~~ first two characters can be anything 3rd position should be a.

'_a%' → first character can start with anything, second position should be a.

'%a%' → 'a' character can be anywhere in word.

write a syntax to get all records which contains atleast 'a'

Select * from T1 where charindex('a', Name) > 0'

order by clause — It sorts the data.

Select * from T1 where id >, 100 orderby id ASC

From
where
Select

Select * from T1 ~~where id~~ order by date DESC

20/1/2015
22/5/2018
23/6/2019

⇒ orderby dg

23/6/2019
22/5/2018
20/1/2015

NOTE :-
In case of order by clause we can use Alias columns.

70

Scanned by CamScanner

Select Row-number () over(order by id Asc) As Rno, from T, order by Rno.desc

From
select
orderby.

To get top 10 rows → Select top(10) * from T order by id

To get middle → Select * from T, order by id Asc
Offset 3 Rows
Fetch Next 3 Rows only.

From
Select
order by
offset.
Fetch.

⇒ orderby clause is mandatory for offset condition and Rank condition

To get bottom rows → Select top(10) * from orderby id ~~desc~~ desc.

Null functions:-

1. is NULL 2. NULLif 3. Coalesce.

is NULL: - It will accept only two arguments, wherever it finds NULLS it replaces with value which we have provided in the second argument.

Select E.Ename as Employee name, is Null (M.ename, 'No manager') as manager name from Employee E Left join Employee M ON E.Mid = M.EID.

O/P	EName	Manager name
	Karthik	No manager
	Paarth	Karthik
	Rithvik	Paarth.

Scanned by CamScanner

T_1	T_2
Id_1	Id_2
NULL	NULL
NULL	NULL
NULL	NULL

Select * from T_1 join T_2 on

isNULL($T_1 \cdot id_1$, 0) = isNULL($T_2 \cdot id_2$, 0)

where isNULL(id_1, 0) = 0 (or)

⇒ In the backend NULL will be taken as 0 and compares with 0 but result will be NULL's only.

⇒ we can keep column names, values anything in isnull arguments

Id_1	Id_2
1	NULL
NULL	2
1	1
2	1
NULL	3
NULL	NULL

Select isNULL(id_1, id_2) from T_1

1
2
1
2
3
NULL

⇒ Select isNULL (1, NULL) = 1

Select isNULL(NULL, 1) = 1

Select is NULL(1, 1) = 1

Sect isNULL(NULL, NULL) = NULL

Coalesce:— It will accept more than two arguments

Syntax:- Select coalesce (exp1, exp2, exp3, ---n) from Tablename

Ex: Select coalesce (NULL, NULL, 5, NULL, 7) o/p: 5

⇒ It will give first non NULL value it will take out of all arguments

Scanned by CamScanner

Select (NULL, NULL, 7) => 7

Select coalesce (NULL, 7, NULL) => 7

Ex:

Id	IName	ICost	CGST	SGST	IGST
1.	Biryani	250	0.05	NULL	NULL
2.	Lollypop	225	NULL	0.05	NULL
3.	cutlet	275	NULL	0.10	NULL
4.	Mushroom	150	0.05	NULL	NULL
5.	~~Paneer~~ Tikka	850	NULL	NULL	0.10
6.	Roti	30	0.05	0.05	NULL

Select Id, IName, ICost, ICost * coalesce (CGST, SGST, IGST) As totaltax from items

union ALL

Select Id, IName, Icost, (ICost * CGST) + (ICost * CGST) As Total tax from items where (CGST is NOT NULL and SGST is NOTNULL)

Nullif:-

Select Nullif(2, 2) = NULL

Select Nullif(2, NULL) = 2

Select Nullif(2, 1) = 2

Select Nullif(NULL, 2) = ~~NULL~~ Error

Select Nullif(NULL, NULL) = ~~NULL~~ Error

First argument of null if cannot be null.

73

Nullif will accept two arguments if both arguments are Same it will give NULL value, if both arguments are Same it will

Scanned by CamScanner

Show nullvalue, first argument shouldnot be null if it is Null it will throw an error.

PID,	pname	PA·C	PSP
1	A	10	10
2	B	20	20
3	C	30	NULL
4	D	40	NULL

Select * from Table where Nullif(pcost, psoldcost) is NULL.

⇒ Query:- 'An apple a day' → only one space should be there b/w two words.

Declare @string varchar(10) = 'An apple a day'.

Select Replace(Replace(Replace(@string, ' ', '12'), '21', '-'), '12', ')

→ An)21212121212apple;

→

2u112

Aggregate functions: — In database management, aggregate is a function where the values of multiple rows are ~~raise~~ grouped together into a single result based on certain criteria.

1. Min 2. Max 3. Sum 4. Avg 5. count.

Min =

pid	pnam	pcost
1	a	10
2	b	20
3	c	5
4	d	15
5	a	7
6	b	18

Select Min(pcost) As minvalue from T_1 ⇒ 5

Select max(pcost) as maxvalue from T_1 ⇒ 20

Select Sum(pcost) as ~~fartal~~ total value from T_1 ⇒ 75

Select Avg(pcost) as avgvalue from T ⇒ $\frac{75}{6}$ ⇒ 12

Count:-

Select count (*) As totalcost from T_1 → It will give null records also

(or)

Select count (pid) as totalcost from T_1 → It doesn't gives null records

(or)

Select count (1) as totalcost from T_1 → It ~~doesn't~~ gives null records

Note:-

1. Any aggregate function will allow single column.
2. Except count ~~for~~ all other aggregate functions can apply on Integers columns only.
3. count (*) ~~& count(1)~~ will give whole records from a table including nulls. where as count(columnname) ~~& count(1)~~ will ignore the null values based upon column which we have specified.

75

Scanned by CamScanner

Select count (*) = 1

Select count (1) = 1

Select count (100) = 1

} count of anything is 1

Select sum (1+5-7) = -1

Select max(7*1) = 7

Select min(1, 100, 1000) = Error → It doesn't allow two (or) more columns. (It requires only one argument).

Select sum(1*7*7) = 49

Select $ = 0.00

Select count (*) + count(*) = 2

Select 1/0 = Divided by zero error ⇒ To avoid this error we have to use nullif ⇒ Select (1/Nullif (0,0)) as value.

Select 3/2 = 1

Conversion functions - To convert from one datatype to another datatype

1. convert
2. cast

2012: Try-convert, try-parse.

Select convert datatype, Expression, styleformat) → will be only used in date formats.

Select cast (Expression as datatype) → no styleformat

Ex Select convert(float, 3/2) = 1.5

Select cast (3/2 as float) = 1.5

Select Avg (convert(float, price)) as avg from Table.

⇒ cast will be used for migration ~~function~~ projects.

cast will be used when we want migrate from one version to another version. It is capable with any of the versions.

Scanned by CamScanner

Declare @date = '01/01/2009'
varchar(10)

Select Convert (Date, @Date, 108) = 2009-01-01.

Select format (Getdate(), 'dd-MM-yyyy')

Try-convert & Try-parse:-

→ This will be used in debugging.

cost (varchar)
100.00
200.05
100,15
15,200,
10.05

T_2 (INT)

Insert into T_2 Select try-convert (float, cost) from T_1 → throws error cannot convert varchar to float

Select cost, try-convert (float, cost) as a from T

where try-convert (float, cost) is NULL

100.00	100.00
200.05	200.05
100, 15	NULL

we cannot use alias functions in where so we have to go derived (or) whole statement we have to write like

Poor sequel

If Else condition:-

In normal, data entry people will enter everything as varchar so it allows ',', while converting varchar to float, if there are 1000 rows which we can't find ',' in that, so by using try-convert use try-convert keeps all errors as nulls, we can filter that nulls by using where condition.

```
Declare @a int = 19

If (@a % 2) = 0
Begin
print convert(varchar(100), @a) + 'is even value'.
End
else
begin
print convert(varchar(10), @) + 'is odd number'.
End.
```


Scanned by CamScanner

Check Condition! –

If (Select vagelimit from voterlist) ≥ 18

Begin
print voter is Eligible
End
else
Begin
print voter is not Eligible
End.

Note!. In if else condition always one of the condition will work either (if) or (else)

→ It is not mandatory to use else condition when we are writing if condition.

→ Incase of if statement we can call select (or) variable (or) a value to write the condition.

→ If our select statements returns more than single value then our operator should be >, ≥, <, ≤, in

→ If it returns single value then "=" operator is sufficient.

If Else :-

Declare @ marks int = 45

If (@marks) ≥ 90
Begin
print 'Distinction'
End
else If (@marks) ≥ 70 and (@marks) < 90
Begin
print '1st class A Grade'.
End.

Scanned by CamScanner

```
else
If (@marks) ≥ 50 and (@marks) < 70
Begin
  print 'B' Grade
End.
else if (@marks) ≥ 30 and (@marks) < 50
Begin
  print 'C' Grade
End.
Else
Begin
  print 'Fail'
End.
```

write a Query to find max Salary in each department.

Employee

⇒

Eid	Ename	Esal	DID
1	A	1000	101
2	B	2500	101
3	C	2700	102
4	D	3000	102
5	E	2500	102
6	F	1750	103
7	G	2750	103
8	H	3250	103

Query:— in department wise

write a ques select highest sal from employee table.

Select max(esal), did from employee group by did.

101	2500
102	3000
103	3250.

⇒ First it groups all similar values and gives the highest of it.

⇒ we can'tget all details in a single groupby statement.

write a query to findout no. of employees in each ~~table~~ department

Select count (ename), did from employee groupby did.

o/p:

Totalcount	DID.
2	101
3	102
3	103

Scanned by CamScanner

Write a Query to get DID which are having more than two employees

O/P: 3 | 101

Select count(*) as a, did b from Employee groupby did having count(*)>2

Sol

⇒ Differences b/w where & having:-

where	Having
1. where clause will be used to filter on top of detailed data	1. Having will be used to filter the aggregated data
2. Groupby clause is not mandatory.	2. Groupby is mandorty for having clause.
3. In where clause first it will filter the data then it will group the columns.	3. First it will group the column then it will perform having clause
4. we cannot use alias columns	4. we cannot use alias columns
5. we cannot use aggregated columns in where clause.	5. we cannot use detailed columns in having clause

⇒ Select count(*) as count2, DID from Employee where did <> 102

Groupby did having count(*) ≥ 2

count	did
3	103

Scanned by CamScanner

CTE:- common table Expression:-

→ CTE will store the data temporarily in memory and when we do any DML operations on top of CTE it will directly affect on the basetable aswell.

Note:-

with in a CTE, we can write 32CTE's, the scope of CTE will be till that execution only.

→ while writing a CTE and after closing the bracket the immediate select (or) Insert (or) update (or) delete statement should be belongs to our CTE

Syntax:-

```
; with CTE1
As
(
Select * from A
)
Select * from CTE1
```

Multiple CTE's

```
; with CTE1
As
(
Select * from A
), CTE2 AS
(
Select * from B
)
Select * from CTE1 C1 join CTE2 C2 ON C1.id=C2.id
```

How to call CTE within a CTE:-

```
; with CTE1
As
(
Select * from A
), CTE2 As
(
Select * from B join CTE1 ON B.id= CTE1.id
)
Select * from CTE2
```

when we call a CTE with in another CTE then it will be known as recursive CTE.

Scanned by CamScanner

Limitations:-

when we are writing a join query inside a CTE then we cannot perform following DML operations i.e insert, update, select.

Imp.

Write a query to remove duplicates from a table using CTE.

Select Row_number() over(partion by Name order by id Asc) As Rno ~~from pro~~

; with deleting duplicates.

As

(

Select *, Row_number() over (Partition by pid, pname, pcost order by pid Asc) As Rno from products

)

Delete from deleting duplicates where ~~Rno~~ Rno <> 1

Product TL

Rno	Pid	Pname	Pcost	Rno
1	1	A	100 ①	
2	1	A	100 ②	
3	1	A	100 ③	
4	2	B	200 ①	
5	2	B	200 ②	
6	3	C	300 ①	
7	4	D	400 ①	
8	4	D	400 ②	
9	1	A	100 ④	

o/p:

Rno	pid	pname	pcost
1	1	A	100
2	2	B	200
3	3	C	300
4	4	D	400.

From partition order by

Select

write a query to get second highest salary in each department.

Eid	Ename	ESal	Did	DR
1	A	100	101	1
2	B	100	101	1
3	C	200	101	2
4	D	200	101	2
5	E	300	102	1
6	F	300	103	1
7	G	300	104	1
8	H	300	104	2
9	I	400	105	1

~~Query~~

Se DR ~~Eid Ename~~ Esal Did

Query:-

```
with T1
As
(
Select *, Dense_Rank() over(partition
by Did orderby Esal desc) As DR
from T1
)
Select * from T where DR=2
```

~~Select Row_n~~
~~; with second highest salary~~
~~(As~~
~~Select*, Row-number over(partion by Did order by Esal desc) As~~
~~Rno from T1~~
~~)~~
~~Delete from secondhighest salary where max(Esal)!=2~~

---

highest → desc
Lowest → Asc.

~~; with Second products~~
~~As~~
~~(~~
~~Select *, Row-number over(partion by Did order by Esal desc) As~~
~~Rno from T1~~
~~)~~
~~(Delete from products where max(Esal)!=2)~~

(or)

Select * from products where ~~max(esal)~~ Rno =2
eid, ename, esal

; with T_1
As
(Select *, Dense_Rank() over (partition by Did order by Esal desc)
as DR from T_1
)
Select Eid, Ename, Esal, did from T_1 where DR = 2.

O/P

Eid	Ename	Esal	Did
3	C	200	101
4	D	200	101
8	H	300	104

⇒

Eid	Ename	EHD	Did	Rno			DR
1	A	20/5/2012	101	1	3		3
2	B	25/7/2012	101	2	2		2
3	C	16/8/2014	101	3	1		1
4	D	10/11/2016	102	2	1		1
5	E	4/5/2011	102	1	2		2
6	F	8/9/2013	103	1	1	3	3
7	G	30/11/2015	103	2	2	1	1
8	I	2/2/2014	104	3	2	1	2

write a Query to get all the employee details in each department who joined recently.

; with T_1
As
(

Select *, ~~Row~~ Row_number() over (partition by Did order by EHD desc) as Rno from T1

) EId, Ename, EHD, DID.
Select * from T1 where Rno = 1

O/p:- EID Ename EHD DID.

3	C	16/8/2014	101
4	D	10/11/2016	102
67	G	30/11/2015	103.

Date functions:-

① Datename
② Datepart
③ Dateadd
④ Datediff.

Day - D (or) dd
Month - MM
year - yyyy
Minute - mm
hour - hh
Seconds - tt (or) ss
Days of week - DW

Datepart:-

S	M	T	W	Th	F	S
1	2	3	4	5	6	7.

Datepart (Day, Expression)

Ex
Select Datepart (~~Day~~ dw, getdate()) → o/p = 2

Select Date name (~~Day~~ dw, get date()) → o/p = Monday.

Select Datepart (~~Day~~ month, get date()) → o/p = 12

Select Datename (month, getdate()) → o/p = December.

Scanned by CamScanner

Date add:—

Select date add (Day, Expression, value)

Select date add (Month, -2, getdate(), -2) o/p => 26/10/2018 (or) 2018-10-26

Date diff:—

Select datediff (day, Startdate, enddate)

Prerequisity- Startdate > End date

Select datediff(Month, '2009-25-07', '2008-25-10') => 9 months.

Select datediff(year, '2018-05-20', '2014-05-20') => 4 years.

(Eomonth) => end of the month.

Select Eomonth(Expression)

Select Eomonth(Getdate()) o/p:- 31/12/2018

Isdate — to know whether date is valid/invalid

Select Isdate (Expression)

Select Isdate (getdate()) => 1

Select Isdate ('02-31-2018') => 0

27/12

Declare @date date 12/31/2018

Set @date = getdate(), 01/01/2019

Query:- To get lastmonth → Nextmonth 1st day.

Select Dateadd (day, 1, Eomonth(@date)) => 01/01/2018

Lastmonth 1st day-

Select Dateadd(Month, -2, Eomonth(@date)) => Eomonth(@date)+1

Scanned by CamScanner

Select dateadd(day, 1, Eomonth(dateadd(~~month~~ day, -2, @date))) m/d/y

⇒ Declare @date date

Set @date = '10/28/2018' → 01/29/2018 → 01/01/2018

write a Query to find out whether the given date is end of the month (or) not.

```
If datepart(Month, @date) = datepart(Month, dateadd(day, 1, @date))
Begin
    print 'given date is not end of month'
end
else
Begin print 'givendate is end of month'
end
```

⇒ write a query to findout max value from below table.

Id_1	Id_2	Id_3	Id_4	Id_5
1	5	15	20	25
26	27	3	4	2
21	29	28	30	31
102	37	38	101	103
99	107	108	102	98

```
Select max(A.maxId) from
(
    Select max(Id) as maxId from T1          ⇒ 102
    union all
    Select max(Id1) as maxId1 from T1        ⇒ 107
    union all
    Select max(Id2) as maxId2 from T1        ⇒ 108
    union all
    Select max(Id3) as maxId3 from T1
    union all
    Select max(Id4) as maxId4 from T1        ⇒ 102
    union all
    Select max(Id5) as maxId5 from T1        ⇒ 103
) A
```

```
Select max(A.max(Id1))
(
    Select max(Id1) as maxId1 from T1
    union all
    Select max(Id2) as maxId2 from T1
    union all
    Select max(Id3) as maxId3 from T1
    union all
    Select max(Id4) as maxId4 from T1
    union all
    Select max(Id5) as maxId5 from T1
) A
```

Scanned by CamScanner

=)

Rno	Bid	from	to	departure time	Arrival time
1	B_1	Chittor	Tirupathi	12/01/2018 05:00:00	12/01/2018 : 07:00:00
2	B_1	Tirupathi	Nellore	12/01/2018 : 08:00:00	12/01/2018 : 11:00:00
3	B_1	Nellore	vijw tin	12/01/2018 : 13:00:00	12/01/2018 : 18:00:00
1	B_1	chem Vellore		12/02/2018 : 04:00:00	12/02/2018 : 06:00:00.
1	B_2	Banglore	Chennai	12/01/2018 : 06:00:00	12/01/2018 : 12:00:00
2	B_2	Chennai	Nellore	12/01/2018 : 00:00:00	12/01/2018 : 05:00:00
1	B_3	Banglore	Mumbai	12/01/2018 : 01:00:00	12/2/2018 : 12:00:00
1	B_4	Banglore	Bangarpet	12/1/2018 : 04:00:00	12/1/2018 : 06:00:00
2	B_4	Bangarpet	Kuppam	12/1/2018 : 07:00:00	12/1/2018 : 8:00:00

Write a Query to findout Bus and departure data wise which bus is departed for first.

```
;with CTE AS
(
Select *, Row_number over(partition by Bid, convert(date, departure time)
  order by departure time Asc) As Rno, * from Bustimings.
)
Select * from CTE where Rno=1
```

⇒ write a Query to findout Buswise which bus is arrived last.

```
;with CTE As
(
Select *, Row_number over(partition by Bid order by arrival time des
as Rno, * from Bustimings
)
Select * from CTE where Rno = 1
```

Scanned by CamScanner

A		B	
Id	Name	Id_1	Name
1	A	1	Null
2	B	2	Null
3	C	3	Null
4	D	4	Null
5	E	5	Null

Query:-

How to update table B from table A.

Update B set Name=A·Name
from A join B on A·Id = B·Id_1.

Query:

Select A·A from
(
Select 1 A, 2 B union all Select 1 A, 3 B) A join (Select 1 A union
Select 1 B) B on A·A = B·A

Sol:

A	
A	B
1	2

B	
A	B
1	3

A·A	A	B
	1	2
	1	3

A	B
1	1
1	1

A	B
1	1

union all

A·A
1
1

A	B
1	1
1	1

A	B
1	1
1	1
1	2
1	3

B
1

Query:

T_1	T_2
Id	Id_2
1	6
2	7
3	8
4	9
5	10

Select $A·Id_1$, $B·Id_2$ from
(
(Select * from T_1) A
Join
(Select Row_number() over (order by Id_2) as Rno, *
from T_2) B ON
join
$A·ID_1$ = B·Rno
)

Scanned by CamScanner

Team T_1	Team T_2			
~~Aus~~ B	~~Aus~~ B	1	1	Aus Ind
Ind	Ind	2	2	Aus pak
Pak	pak	3	3	Ind pak

as scheduled

```
Select A.team + 'vs' + B.team from
(
(Select Row_number() over(order by Team Asc) as Rno1, * from T1) A
~~Left~~ join
(Select Row-number() over(order by Team Asc) as Rno2, * from T2) B

        on A.Rno1 > B.Rno2)
```

O/P

AUS vs IND

AUS vs pak

Ind vs pak

Query 2.

```
Declare @ Date varchar(100)

Set @ date = '01/01/2018'
             '01/01/2018'
             '1/1/2018'
             '01/1/2018'
             '01/01/18'
             '1/1/18'

Select
case when left(@date, charindex('/', @date) -1) = 1
then '0' + left(@date, charindex('/', @date) -1))
else
left (@date, charindex('/', @date)-1
+ '/'
```

To

Scanned by CamScanner

28/12/18

while loop:-

Write a query to print first 10 numbers.

```
Declare @Start int, @end int

    Set @ Start = 1

    Set @ end = 10

while( @start ≤ @end)

  Begin

   print @Start

    Set @ Start = @Start+1

    End.
```

=) write a Query to print first top 5 even numbers.

```
    Declare @start int, @end int

      Set @ Start=1

      Set @ end =10

  while (@start ≤ @end)

  Begin.

  ~~Begin~~ If ~~Mod~~( @start % 2)=0

      Begin
        print convert(varchar(100), @start + 'is even number')

      end

    Set @ start = @ start+1

  else
  Begin print convert (varchar(100), @start + 'is odd number')
  end.
```

Scanned by CamScanner

Eid First Last middle emailId.

1. Parth Kumar Samthan

2. Dhruv Chatt Kumar
arjee

3. Madhuri Telug Naidu
uti

4. Vishali Telugati Roy

Declare @String varchar(10)

Declare @min @ int @max int

~~Select~~ Set @ min = (Select (min(eid) from T1) → 1

Set @max = (Select (max(eid) from T1) → 5

while (@min ≤ @max)

Begin

update T1 set emailId = Last + First + middle + convert(varchar(10), @min)
+ '@gmail.com' where eid = @min.

set @ min = @min + 1

end

Concat:-

Syntax:- Select concat (col1, col2, col3) from T (exp1, exp2, exp3)

Note:
concat will combine all integers & varchar values into a string.

Scanned by CamScanner

G	o/p
M	F
M	F
F	M
F	M
M	F

Query:- To print F in place of M & M in place of F.

Select case when gender = 'M' then 'F' else 'M' end from T

(or)

Select (case when gender = 'M' then 'F' when gender = 'F' then 'M' end) as Gender from T

Case statement:-

Case (i):- Single case with single else

Case when col1 = 'value1', then col2 = 'value2' else col3 = 'value3' end from T.

Case (ii):- Single case with multiple when

Select case when col1 = 'value1' then 'value2' when col2 = 'value3' then 'value3' end from T

Case (iii):- Multiple Cases

Select case when col1 = 'value1' then 'value2' end as col1, case when col2 = 'value3' then 'value4' end as col2 from T

Note:-

1. Each case statement may (or) may not have else statement.
2. Each case should end with 'END' keyword.
3. In a case statement we can write 'N' no. of "when" conditions.
4. In a ~~case~~ when statement we can keep ~~multiple~~ same columns (or) values by using (and (or) or) all operators.

Select case when col1 = 'value1' and col2 = 'value2' and col3 = 'value3'.

Scanned by CamScanner

Query: To print sum of positive and negative values in two columns.

T1

Id.	pos	Neg
-1	15	-15
-2		
-3		
-4		
-5		
5		
4		
3		
2		
1		

Sol. Select Sum(case when id<0 then id end) as negative

Sum(case when id>0 then id end) as positive

Query:

```
Declare @date varchar(100) = '01/1/18'.

Select case when left(@date, charindex('/', @date) -1) =1 then '0'+
left(@date, charindex('/', @date)-1)
else
left(@date, charindex('/', @date)-1) + '/' +
case when substring(@date, charindex('/', @date, charindex('/', @date)+1)
-1, charindex('/', @date, charindex('/', @date)+1) - charindex('/', @date)
-1) =1 then '0' + substring(@date, charindex('/', @date, charindex('/', @date)
+1)-1, charindex('/', @date, charindex('/', @date)+1) - charindex('/', @date)
-1
else
Substring(@date, charindex('/', @date, charindex('/', @date)+1)-1,
charindex('/', @date, charindex('/', @date)+1) - charindex('/', @date)
-1
end + '/'+
```

Scanned by CamScanner

case when Len(Right(@date, charindex('/', @date) - 1)) = 2 then '20' + right(@date, charindex('/', @date) - 1) else right(@date, charindex('/', @date) - 1)

end.

Scanned by CamScanner

T-SQL (Transaction SQL)

- View
  - System defined views (predefine views) (SQO → DBA)
    - Information Schema views
    - Catalog views
    - Dynamic Management views
  - User defined views.
    - Simple view
    - complex view
    - Index view
    - check option views.

View: view is nothing but a virtual object (or) a mirror object and also it Saves the Sequel query.

Note:- By using the views we can provide the column level Security

→ view doesn't store any data

→ while creating a view we should keep select statement inside the view

→ Incase of simple view we can able to perform DML operation on top it and it will directly effect on the physical table data.

→ Incase of using aggregate function (or) calculated columns we can't able perform DML operations on top of views.

Eid	Esal	DID
1	100	101
2	200	101
3	300	102
4	400	102

Did	Sal
101	300
102	100

```
Create View viewname
As
(
Select sum(sal), did from T
group by did
)
```

we cannot update Sal value in main table we have detailed info and in aggregated table we have combined date

Scanned by CamScanner

update view set sal = 500 where did = 101 and sal = 300 → It is not applicable.

Syntax for creating a views:— select * from Information schema. columns

select * from Information schema. tables → No. of columns in database table

↓

Information schema Referential constraints

create view viewname

As

(

select col1, col2, col3 from T_1

)

Select * from T_1

Note:-

We cannot call temporary tables & table variables inside the view

⇒ we cannot call userdefined functions inside a view:(

Simple view:—

⇒ If we create a view on top of single table then it is known as "simple view"

Syntax:-

create view sv-Employee

As

(

Select Eid, Ename, Esal from Employee

)

⇒ If we drop any column in physical table, if we execute the view, it will throw an error if we refresh also, if we want then we have to drop the view & then recreate.

Syntax to retrieve the data from simple view:-

Select * from sv-Employee.

Simple views can not be updatable ~~only~~ incase of ~~not~~ using aggregate & calculated columns.

Alter view SV-Employee

As

(

Select col1, col2, col3 from Employee

)

} to get column also in view.

⇒ If we delete column from maintable then it will throw error while executing view.

Scanned by CamScanner

=> If we change the structure of table (or) ~~datatype~~ column (or) Table names (or) columnnames existing view will throw an error, we can rectify that error by using alter statement.

=> If we change the datatype of particular column then it won't directly reflect on view until unless we refresh that view.

Syntax:- Refreshing a single view.

sp - Refreshview 'viewname'.

=> Refreshing All views

sp - Refresh view all

Syntax to drop view:-

Drop view viewname

we can create view for other database objects.

Complex view:-

→ If we create a view for more than one table then it will be known as complex view.

```
create view cv-Emp-dept
As
(
Select E.Empid, E.Esal, E.Ename from Employee E join dept d on e.did=d
)
Select * from cv-Empt_dept.

update cv-Empt-dept set sal=1500 where dept id = 'IT' → we cannot updati
```

=> we cannot use DML operations on top of complex views.

=> we cannot drop the columns in views because it doesn't store any data.

~~we cannot insert in views. becoz if we insert values in views it cannot update data in original table~~

Scanned by CamScanner

Index view:-

If we create a view on top of a table with schema binding then it will be known as "schema binding view."

2. on top of schema binding view if we create unique nonclustered index then that unique will be called as indexview (or) materalised view but directly we cannot create unique nonclustered index so first create unique clustered index and then go with unq

3. Indexview can store the data after creating unique nonclustered Index

4. when we create schema binding view we ~~cannot~~ cannot/can able to change the Schema of that ~~view~~ table/view (structure)

⇒ we cannot drop the physical table in indexview.

5. while creating schema binding view, it is mandatory to mention the all columns name instead of '*' and schema name before the table.

6. we ~~cannot~~ can perform DML operations on indexview.

(F)

7. If we want to perform DML operations on top of index view then create "instead of trigger" on top of that view then perform the DML operations.

8. If we dump the data then constraints will not come.

8. To check the memory used by any object ⇒ sp_Spaceused 'objname'.

Syntax:

create view 'IV_viewname'
with schema binding
as
Select Eid, Ename, Esal from Employee
)

① create unique clustered Index. uCI on IV_viewname (Eid)

② create unique nonclustered Index unCI on IV_viewname (Eid).

Scanned by CamScanner

Check option view:—

Create view cov_Employee as
(
Select * from Employee where id<7 with check option;
)

⇒ we can perform DDL & DML operations.

Synonyms:—

→ It's an alternative name for existing table.

Syntax:

create synonym synonymname for oldtable.

DB_1	DB_2	DB_3
Flipkart → Koramang	Flipkart → Btm	Flipkart → HSR

Tbl_Gstrates

Pro	Gstrate
1	X
2	Y
3	Z
4	A
5	B

create synonym sy_GsTrates
For DB_1.DBO.Tbl_GsTrates
Select * from sy_GsTrates.

create synonym sy_GsTrates
For DB_2.DBO.Tbl_GsTrates
Select * from sy_GsTrates

Note:-

→ when we perform DML operations on top of synonym it will directly reflect the changes in physical table and viceversa.

→ If we change the tablename, we have to drop the synonym and then recreate synonym.

→ whatever the properties we have for physical table will be applicable for synonym also like constraints, identity etc.

Scanned by CamScanner

=> Synonyms can be created on whole table, cannot create for particular columns.

=> we cannot use alter statement in synonym (there is no alter command to do it).

=> We can perform joins b/w synonym to synonym, synonym to view, synonym to temp table, view to temptable.

=> we cannot create synonyms for temporary objects.

Can we create a synonym on top of view (or) not → No

Cursors:- when we need to update records in a table in one step i.e row by row.

① Declare cr_updateemail cursor for select Eid, Efirstname, Emiddlename, Elastname from Employee

② open cr_updateemail

Declare @Eid int, @EFirstname varchar(10), @middlename varchar(10), @Elastname varchar(10) → we have to declare the variables for that columns.

③ Fetch next from cr_update email into @Eid, @EFirstname, @Emiddlename, @Elastname

while @@Fetch_Status = 0

Begin

update Employee Set EmailId = concat(@EFirstname + @Emiddlename + @lastname + 'gmail.com') where Eid = @Eid

Fetch next from cr_update email into @Eid, @Firstname, @Emiddlename, @Elastname.

End

④ close cr_update email

⑤ deallocate cr_update email.

Scanned by CamScanner

⇒ cursor will do row by row operation and we have to follow the following five steps while declaring a cursor.

1. Declare
2. open
3. Fetch
4. close
5. Deallocate.

Note:-

The initial fetch statement is used to fetch the first row only

⇒ The second fetch statement is used to fetch from second row to till end

⇒ It is mandatory to close & deallocate the cursor

⇒ when one cursor is in open state we cannot declare the another cursor in same session.

⇒ The scope of cursor will be till that execution only

Slow Execution ⇒ Hit the table & storing it in variables.

⇒ It has to hit the table every time & store it in variables.

To print evennumbers in a table:-

Declare cr_evennumbers cursor for select @Eid from Employee.

open cr_evennumbers

Declare @Eid int

Fetch next from cr_evennumbers into @Eid

while @@fetch_status = 0

Begin

Scanned by CamScanner

```
if (@eid % 2) = 0
Begin
print @eid
end
fetch next from cr-evennumbers into @eid
end
close cr-evennumbers
deallocate cr- evennumbers.
```

5/1/2018

Sub query & co-related subquery:-

Subquery contains inner and outer queries, when we execute a subquery first inner query will get executed and based upon that result set outer query will get executed.

→ Inside a Subquery we can call 32 queries.

Syntax:

Select * from A where col1 in (select * from B) ⇒ Two

outerquery ② col2 Inner Query ①

Select * from A where col1 in (select col2 from B where col3 > (select col3 from C))

Ex:

Id_1	Id_2
2	1
3	5
4	6
5	7
	8

Q) Integers which are present in A but not in B.

① Select * from A where Id_1 not in (select Id_2 from B)

② Except

③ Left join by removing nulls.

o/p: 2
3
4
5

Scanned by CamScanner

Except:

Select Id_1 from A

Except

Select Id_2 from B.

Left join:-

Select * from A left join B on $A.Id_1 = B.Id_2$ where $B.Id_2$ is null

Note:-

⇒ In case of subquery we should not use orderby clause in innerquery until unless we use top clause.

⇒ If we are using "=" operator then inner Query will give only one record, if it is giving more than one value then it will throw an error saying that subquery is returning more than one value.

⇒ To allow multi values we have to use <, >, not in, in operators.

Syntax to compare the latest record in B table:-

Select * from A where col1 (Id_1) = (select col_2 top(1) from B orderby Id desc)

write a query to findout the records which are greater than B table and not in C table

A	B	C	AZ
Id_1	Id_2	Id_3	
1	1	7	40
2	2	8	
10	5	9	
20	7	10	
30	30		
40			

B not in C

2

5

30

104

Scanned by CamScanner

Select * from A where Id_1 > any (Select Id_2 from B where Id_3 not in (Select Id_3 from C)

o/p: 10, 20, 30, 40.

Corelated Subquery:-

write a query to find employee information whose Salary is greater than their average Salary in each department.

Select * from Emp E_1 where Sal > any (Select avg(sal) from Emp E_2 where E_1.did = E_2.did)

Eid	Ename	Esal	Did	~~Eid~~	~~Ename~~	Esal	did
1	A	100	101			100	101
2	B	200	101			200	101
3	C	300	102			300	102
4	D	400	102			400	102
5	E	500	102			500	102

write a query to find employee information whose Salary is greater than their average Salary

Select * from Emp E_1 where Sal > any (Select avg(sal) from Emp E_2)

Merge:-

By using merge statement we can perform three DML operations at a time i.e Insert, update, delete.

105

Scanned by CamScanner

⇒ Merge statement is required a key column to identify matching and unmatching records.

⇒ Merge statement should always end with ';'

Syntax:-

Merge into targettable t using sourcetable s on t.column = s.column

when matched then update set target.col = s.col

when notmatched then insert target column list values Source column list values

when notmatched by source then delete output & action inserted.eid, deleted.d into #temp.

write a syntax to dump the data from temptable to physical table

Source temp				dest phy			
Pid	Pname	pcost	pgst	pid	pname	pcost	pgst
1	A	100	0.10				
2	B	200	0.20				
3	C	300	0.30				
4	D	400	0.40				
5	E	500	0.10				

Step 1:-

Merge into product ~~table~~ as p using #product as p1 on p.pid = p1.pid

when matched then update set pgst = p.pgst , pcost = A.pcost

when not matched then insert pid, pname, pcost, pgst values (p.pid, ppname, p.pcost, p.pgst);

106

Scanned by CamScanner

dest

pid	pname	pcost	pgst
1	A	100	0.10
2	B	200	0.20
3	C	300	0.30
4	D	400	0.40
5	E	500	0.10

Step2:-

Insert into #Product values (6, 'F', 600, 0.70)

update #product set pgst = 0.30 where pid = 1

⇒ Execute the merge statement

#product ⇒ source

pid	pname	pcost	pgst
1	A	100	0.30 → updated value.
2	B	200	0.20
3	C	300	0.30
4	D	400	0.40
5	E	500	0.10
6	F	600	0.70 → New record

Step3:

Insert into product values (7, 'H', 700, 0.80)

product ⇒ destination.

pid	pname	pcost	pgst
1	A	100	0.10
2	B	200	0.20
3	C	300	0.30
4	D	400	0.40
5	E	500	0.50
7	H	700	0.80

⇒ Execute the merge statement

⇒ After executing, unmatched record from destination will be get deleted.

→ In source, new record is not there so it will be get deleted

Scanned by CamScanner

write a delete statement to delete the records which are present in A but not present in B without leftjoin (or) merge

Delete * from A where Id not in (select Id from B)

Output:

$Action, Inserted.pid,

$Action	pid	pname	pcost	
Inserted	1	a	10	Inserted.pname,
"	2	b	20	Inserted.pcost,
"	3	c	30	Inserted.pid
Deleted	1	a	10	
deleted updated	1	a	20	Deleted.pcost as pcost, Deleted.pname as pcost, # into temp,

Views:-

Catalog views:-

Select * from sys.objects

Store procedures type = S

user defined type = U

=) For Schema dbo schemaID is default

Select * from sys.tables

Select * from sys.columns

Select * from sys.views

Select * from sys.synonyms.

Scanned by CamScanner

Dynamic management views - It shows how much storage amount has given.

Select * from
DM_ exec_ procedure _ stats

Select * from [sys]. [dm_ exec_ query _ stats]

Select * from [sys]. [dm_db_index_usage_stats]

user defined views:-

⇒ when we are

Index views:-

⇒ Index can be created only on top of physically existing objects

- when we create a primary key automatically clustered index will get created and in this case column won't allow null records & duplicates.

⇒ when we create a clustered index manually without primary key then it allows duplicates & nulls. It has sorting property.

⇒ If we want to make the column as distinct then create unique clustered index.

⇒ when we are performing DML operations on checkoption view it will allow only those rows are satisfied with condition otherwise we are unable to perform DML operations.

Scanned by CamScanner

7/1/2018

Triggers

1. DDL Triggers.
2. DML Trigger
3. Instead of trigger
4. Logon trigger.

⇒ Triggers are automatic fired action.

⇒ we are using triggers to maintain audit information and it will be created only on top of few internal tables like clientdetails, employeedetails etc.

For insert trigger:-

```
create trigger triggername
(tr_employee)
on Employee
For Insert
as
Declare @ Eid int, @ Ename varchar(10), @ Esal float, @ did int, @ currentuser
varchar(10), @ date date, @ Servername varchar(10), @action varchar(10)

Select @ Eid = i.Eid from inserted i
Select @ Ename = i.Ename from inserted i
Select @ Esal = i.Esal from inserted i
Select @ did = i.did from inserted i
Set @ currentuser = current_user()
Set @ date = getdate()
Set @ servername = @@ Servername
Set @ action = 'Inserted a new record'
```

Addit table

```
create table tr_Employee
(
Eid int,
Ename varchar(10),
Esal float,
did int,
currentuser varchar,
date date,
Servername varchar,
action varchar(10)
)
```

Scanned by CamScanner

Insert into tr-employee values(@Eid, @Ename, @Esal, @did, @current user, @date, @servername, @action)

Syntax to enable/disable a trigger:-

Alter table tablename
Enable trigger triggername

Alter table tablename
Disable trigger triggername

Delete trigger:-

Create trigger triggername
(tr-employee)
on employee
For delete
as
declare @Eid int, @Ename varchar(10), @Esal float, @did int, @currentuser varchar(10), @date date, @servername varchar(10), @action varchar(10)

Select @Eid = d.Eid from deleted d
Select @Ename = d.Ename from deleted d
Select @Esal = d.Esal from deleted d
select @did = d.did from deleted d
~~Select~~ set @current user = current_user()
set @date = getdate()
set @servername = @@servername
set @action = 'Deleted a record'
Insert into tr-employee values (@Eid, @Ename, @Esal, @did, @currentuser, @date, @servername, @action).

Scanned by CamScanner

update trigger :-

```
create trigger tr-up- Employee
on Employee
for update
as
Declare @Eid int, @Ename varchar(10), @Esal float, @did int, @current user
varchar(10), @date date, @servername varchar(10), @action varchar(10).

Select @Eid = i.Eid from inserted i
Select @Ename = i.Ename from inserted i
Select @ Esal = i.Esal from inserted i
Select @ did = i.did from inserted i
Set @ currentuser = current-user()
Set @ date = getdate()
Set @ servername = @@servername
Set If update(esal)
  Begin
    Set @action = ' Record is updated'
  end
  If update( eid)
  Begin
  Set @action = ' Record is updated'
  end.
Insert into tr-up employee values (@Eid, @Ename, @esal, @did, @ currentuser,
@ date, @ servername, @action).
```

112

Scanned by CamScanner

Instead of trigger —

```
Create trigger tr-transactions
on transaction
Instead of insert
as
Declare @Tid int, @Mname varchar(10), @Tamount float, @currentuser
varchar(10), @date date, @Servername varchar(10), @action varchar(10),
@day varchar(10)
Set @ Tid = i.Tid from inserted i
Set @Mname = i.Mname from inserted i
Set @ Tamount = i.Tamount from inserted i
Set @ currentuser = current_user()
Set @ date = getdate()
Set @ Servername = @@servername
Set @ action =
Set @ day = datename(dw, getdate())
If (@day) in ('saturday', 'sunday')
Begin
Raise error('system is under maintaince process, 16, 1)
Roll back
Set @ action = 'Error record'.
end
else
Begin
Insert into transaction values(@Tid, @Mname, @Tamount, @currentuser,
@date, @Servername, @action, @day)
```

Scanned by CamScanner

commit

Set @action = 'Record is inserted'

end

Insert into tr-transaction (@Eid, @Mname, @Tamount, @currentuser, @date, @servername, @action, @day).

Insert, delete, upate in a single trigger

Create trigger tr_insert_delete-update

on Employee

For insert, delete, update

Declare @Eid int, @Ename varchar(10), @Esal float, @did int, @current user varchar(10), @date date, @servername varchar(10), @action varchar(

Select @Eid = i.Eid from Inserted i

Select @Ename = i.Ename from inserted i

Select @Esal = i.Esal from inserted i

Select @did = i.did from inserted i

Set @current user = current-user

Set @date = getdate()

Set @servername = @@servername

Set @action = 'Inserted a new record'

~~Insert into~~

Scanned by CamScanner

8/1/2018

User defined functions:-

1. Scalar functions.
2. Inline functions:- → Inline table valued functions
   → Multi table valued functions.

Functions:-

→ A function should return a single scalar value in o/p which can be any datatype except image datatype.

Limitations:-

→ A function should return a value in o/p and its mandatory
→ we can pass around 2100 i/p parameters inside a function.
→ we cannot use o/p parameters inside a function but we can use in stored procedures.
→ we cannot perform DML operations in scalar and table valued functions.
→ we can perform DML operations in multi table valued functions.
→ we cannot call temporary tables inside a function
→ we cannot do error handling (try-catch blocks) using functions.
→ Functions won't store executions plans for reusability cases which will degrade the performance.
→ we cannot call stored procedures inside a function.
⇒ we can call function inside a function
⇒ we have to execute the functions by using only select statement
⇒ when we are executing any function it is mandatory to mention schema name eventhough it is a default schema i.e dbo.

Scanned by CamScanner

Scalar functions:-

```
create function fn-add two numbers
(
@a int
@b int
)
Returns int
as Begin
Declare @c int
Set @c = @a + @b
Return @c
end
```

Execution:

```
Select dbo.fn-addtwonumbers(1,2)
```

write a function to get employee count based on did.

EmpID	Ename	ESal	Edid
1	A	100	101
2	B	200	101
3	C	300	102

```
create function fn-employee count
(
@didint
)
Returns int
as
Declare @b int
Set @b = ~~count~~ select count(eid) from employee where did = @did
Return @b
```

Execution

```
Select dbo.fn-employee count(101) from emp where did = 101
```

Scanned by CamScanner

Write a function to get employeename when we are passing Department id and employee id.

```
create function fn_employee name
(
  @ Did int
  @ Eid int
)
 Returns varchar (10)
as      Declare @employee name varchar(10)
 Set
 Declare @ employee name = (Select employeename from table where
                            did= @ did and eid = @eid)
                                  enam
   Return @ employee name
```

Execution: Select dbo.fn_employeename (101, 1)

⇒ we use scalar functions in converting numbers to currency function.

Create a function to retrieve a substring value which occurs b/w 2nd & 3rd underscore. a-b-c

```
create function fn_a_b_c
(
  @ a-b-c varchar(10)
)
 Returns varchar(10)
as
Declare @ String a-b-c varchar(10).
 Set @ String a-b-c = (Select Substring (@ a-b-c, charindex('_', @ a-b-c)+1,
```

Scanned by CamScanner

charindex[('-', @a-b-c, charindex('-', @a-b-c)+1) -

charindex('-', @a-b-c) -1]

Return @string c.

write a function to get employee details depends upon their department ID.

create function fn_employee details

(
@ did int
)

Returns table

as Return(select * from emp where did = @did)

Return R

Table valued:- Inline table valued functions:-

Create function fn_employee details

(
@ did int
)

Returns table

as

Return(Select * from emp where did = @did)

Ex: Select * from dbo.fn_employee details.(101)

Table datatype:- It will store the whole object data, table data.

Scanned by CamScanner

In line
Table valued:-

```
create function fn-productsal
(
@Date date
) Returns table
as
declare @year int
  set @ year = datepart(year, @date)
  Return (Select * from product where year = @year)
```

Note:

In Inlined table valued function we should use only one select statement and we cannot perform DML operations

Multistatement table valued function:- To get data of two tables in each data in one single resultant table

```
create ~~table~~function fn_st-te-data
(
@ Sid int,
@ Tid int
)
Returns @ st_te_data ~~table~~ table
( id int,
  Name varchar(10),
  DOB date
)
As
```

Scanned by CamScanner

insert into @sttedata select * from student where sid=@sid.

Insert into @sttedata select * from student where Tid=@Tid

Return. O/P select * from dbo.fn(1,1)

Drop function functionname → To drop a function.

Alter ~~table~~ function ~~tablename~~ functionname

Add ~~column~~ columnname datatype.

Split function:

@string = 'a,b,c,d,e'

'a|b|c|d|e'

O/P: a
b
c
d
e

120

Stored-procedures:-

Stored-procedures contains set of statements which includes all kind of DML operations.

=> when we are executing a stored procedure for first time it will compile each & every statement and creates execution plan which will be reusable for next executions.

⇒ A Stored procedure will prefer differ name resolution

Differ name Resolution:-

when we are creating store procedures for non existing objects, storeproceduri will gets created becoz while creating a store procedure SQL Server engine will check whether the SQL code (syntaxes) is proper (or) not.

→ while executing the store procedures it will compile each & every statement and then it will throw the error.

This whole process is called as differname resolution.

⇒ Inside a stored procedure we can call all the objects (tables, views, synonyms, temp tables) except triggers.

⇒ Inside a stored procedure we cannot create ~~triggers~~ triggers & views (synonyms)

⇒ we can call functions inside a stored procedure.

⇒ we can call child stored procedure inside a procedure called recursive.

⇒ we can pass around 12,000 I/p & o/p parameters for a stored procedure

⇒ we can use any datatype inside a stored procedure

⇒ we can do error handling by using try catch blocks inside a store procedure.

we can create synonym for Storedprocedure.

Scanned by CamScanner

⇒ Stored procedures will be executed by using Exec command.

⇒ A store procedure may (or) may not return a value.

⇒ we can call 32 child stored procedures inside a stored procedure.

⇒ we can create a stored procedure without begin & end statements as we In programmability we will see sp.

Syntax:-

```
create procedure stp_Get order details
(
@ order id int
)
AS
Begin
Select * from orderdetails where orderid = @orderid.
end
```

Execution: Exec stp-get order details
@ orderid = 1

DB_1	DB_2
P_1	P_2 (Sy)
T_1	
T_2	
T_3	

Select ~~st from~~ sp should be there

Create a procedure to read the ~~data~~ acc details based up on Transaction id

Sol: create procedure stp_accdetails

```
(
@ tid int
)
```

Scanned by CamScanner

as

Begin.

Create table #tempiable.

(

Aid int

Aname varchar(10)

) Tid int

~~Select * from #ter select *.~~

~~Insert into #temp from Accountdetails~~ ~~to~~ h.

Inserer into # temp

Select A.Aid, A.Aname, T.Tid from Accountdetails A join transaction t on

A. Aid = T. Aid where T. Tid = @Tid.

Select * from #temp

end.

Executes Exec ~~Ste~~ account details

~~where~~ @Tid = 1

⇒

Branchno	Branchaddress.	Branchno	docid	dname.
1.	Hyd	1	1.	Thomas
2.	Vjw	1.	2.	Kelvin
3.	Guntur	2	3	Marin.

⇒ Create a procedure to get Branchdetails based on docid.

Create procedure ~~s~~ stp - Branchdetails

(@docid int

) as

Begin

Create table tempbranchdetails

(Brancho int,

Branchaddress varchar(20),

```
docname varchar(10),
docid int
)
Insert into tempbranchdetails
Select B.Branchno, B.Branchaddress, d.docname, d.docid B join d on
        B.Branchno = D.Branchno where B.docid = @docid

Selecct * from tempbranchdetails

end
Exec stp-Branchdetails @docid = 1.
```

Cumulative count:—

```
Select sum(transaction amount) over(orderby TID asc) from #Transaction.
```

```
⇒ Select *, sum(case when transaction type = 'credit' then Transanction
amount else -1 * Transanction amount end) over (order by TID asc)
as accountbalance from #transactions.
```

Calling a child procedure inside a procedure:—

```
After p1
create proc p1
as Begin
Create table #temp
(id int)
exec p2
End
```

```
Create proc p2
as begin
Insert into #temp values (1)
Select * from #temp
end
exec p1.
```

124

Scanned by CamScanner

⇒ Queries.

List of Name starts & ends with vowels which doesn't contains any duplicates

Ename

Aishwarya Anu Anu

Aishwarya Anushka

A)

Select Ename from vowels where Ename like '[Aeiou]%[Aeiou]'

union

Select Ename from vowels where Ename like '[Aeiou]%[Aeiou]'

125

Scanned by CamScanner

18/1/19

SSRS:-

SQL Server Reporting Services:-

1) Table Report
2) Matrix Report
3) Sub Report
4) Drill Down Report
5) Drill through Report
6) Parameterized Report
7) Cascading Report
8) Linked Report
9) Cache Report
10) Snapshot Report
11) Groupings
12) SSRS Expressions
13) Deployem
14) Repor Server
15) Report Manage

SQL Server	visual studio
2008R2	2008R2
2012	2010 | 2012
2014	2013
2016	2015

=> How to open visual studio:-

→ Go to Search

→ Type DEVENV (or) visual studio · click on it.

How to create a project:-

→ Go to visual studio

→ Click on new project

→ Select Reporting Services under BI

→ choose Report Server project

→ provide project name & if required Solution name · click on ok

How to create SSRS Report:-

→ Go to visual studio

click on existing project under Recent.

126

Scanned by CamScanner

→ Go to Solution explorer

→ Right click on reports

→ click on Add

→ Select New item

→ In wizard Select Report option & provide the report name (Make sure that extension should be · RDL)
↓
Report ~~Developement~~ definition language.

→ click on add

→ Go to Report data in right hand side (to get Report data option go to view & click on last option) ctrl+Alt+D.

→ Right click on datasource

→ Click on add datasource.

→ provide datasource name

→ choose the type as Microsoft SQL Server.

→ Go to connection string

→ click on edit

→ provide the Servername

→ choose the authentication.

→ If required provide the username & password

→ Select the database.

→ Click on test connection.

→ If it is succeeded click on OK.

→ If it is a SQL Server authentication

→ Go to credentials tab

→ provide username & password. click on OK.

127

Scanned by CamScanner

→ Right click on Dataset
→ Click on Add Dataset
→ Provide dataset name
→ Choose Embedded dataset
→ Go to data source
→ choose the existing datasource ↓ Query type → Text
→ In Query box, write any Select Query
→ Click on Refresh fields
→ Click on OK.
→ Go to view
→ Click on toolbox
→ Drag table option into design Surface.
→ using data label select the required fields and preview the report.

Data Sources:-

By using datasource, we can connect to a particular Source (Servers) where the data exists. There are two classifications in datasources

1. Embedded data Source.
2. Shared data Sources

→ when we create embedded datasource, it will be used with in the report only.

→ when we create shared datasource, it will be used across all the reports with in a project.

Scanned by CamScanner

Dataset:- By using dataset we can write the Querys and get the data from the objects. It is also classified into 2 types.

1. Embedded dataset
2. Shared dataset

Refresh:-

when we are previewing a report for first time, it will hit the database and get the fresh data and stores it in cache memory.

→ For further execution, it will always brings the data from Cache memory instead of ~~dat~~ bringing the data from database

→ If we want to bring latest data again, we want to press again Refresh button.

Rendering / Exporting options:-

→ For exporting purpose we use

1. word
2. Excel
3. powerpoint
4. PDF
5. TIFF file
6. MHTML (webarchiere)
7. CSV (common delimited)
8. XML file with report data.

⇒ Headers of Report can be changed but data cannot be changed until unless we change it in database.

default → Specific → one name.

distinct → available values → get values from query.

General → prompt → select a value

Scanned by CamScanner

Parameter → It's a filter

1. Query level parameter → In database level, it will filter the data.

2. Report level parameter → Run the query all data will bring from database, after bringing the data, we will keep filter.

1/19/2018

How to create shared datasource & shared dataset.

→ on Right hand side of screen, right click on shared data source

→ Add new datasource

→ ~~Enter~~ choose datasource name from exisiting databases as datasource name

→ choose type as microsoft SQL Server.

→ In connection string, click on edit provide the servername

→ click on windows authentication.

→ Enter database name and click on test connection

→ If test connection is succeeded click on OK.

→ It displays Datasource as servername & database name as initial catalog. & click on OK.

→ Shared data source ends with .rds (report data source) & reports ends with .rdl (report definition language)

→ The backend code for both Rds & Rdl is XML format.

→ Right click on reports and click on viewcode then XML code occurs & close it

→ If we do any changes in XML code then report format also changes.

Scanned by CamScanner

→ click on left datasource, add datasource provide name & select Shared data source reference & select the database name & click on OK.

→ click on dataset, add dataset, click on use a dataset embedded in my report

→ ctr-select datasource & write the query & press Refresh fields

→ Ro press fields then field names will appear & click on OK.

→ In design, Right click on table, Right click on Insert and select table

→ Add columns.

→ click on preview to view the table data.

Parameter:-

→ It is a filter, based upon that our report will filter the data & represents in o/p.

→ It is of two types:

1. Query level parameter
2. Report level parameter.

page No still PDF

1. Query level parameter:- In database level, it will filter the data.

→ Right click on dataset name, ~~double~~ in query level write the condition based query (Ex: Select * from Emp where Name = @Name)

→ In parameters, we can see that parameter name (@name)

→ Double click on parametername, then Report parameter properties window will come

Scanned by CamScanner

→ In report level, we can pass the values before previewing (or) after previewing.

Before preview:-

→ Go to default values

→ click on specify values (passing values manually)

→ click on add and enter the parameter value

→ click on preview, it will whenever we preview parameter value will be passed to query, when it hits the database, there itself it will filter the data and it will fetch the data.

→ whenever we preview it second time, it will fetch the data from cache memory and it doesn't hit the database again.

→ create new dataset

→ Double click on parameter name, Go to available values, if we want to get the values in a drop down format ⇒ Select get values from a query.

→ Select the particular dataset name

→ Select value and label fields & both should be same. click on ok.

→ Go to preview, we will get dropdown option & select the value & click on ViewReport which displays the result.

→ If we want multiple values then change the operator in query in dataset.

→ ~~If we want multi~~

→ click on parameter name, go to general, write prompt heading in and datatype as text.

Allow multiple values, Parameter visibility → visible

Scanned by CamScanner

→ In default values we have to mention any value otherwise it throws error.

→ click on preview then 'Select all' comes, if we select multiple values and click on viewreport then it gives multiple o/p's.

Report level parameters :—

Right click on Reports & Create new Rdl

→ Here we don't pass any parameter values & here we need to create data source & dataset & don't use where condition in Query box.

→ Here we need to create parameter manually

↓

Double click on parameter & click on general.

↓

Provide name, prompt & datatype

↓

Click on OK → Create dataset to drag the table.

→ If we want to give filter on Report level parameter

↓

Right click on tablix

↓

click on tablix properties.

↓

Go to filters & click on Add

↓

In expression choose the column name to filter data & Select the operator & pass values to click on fx

↓

Go to parameter Select value click on ok

↓

click on ok.

Scanned by CamScanner

Preview & Select parameter value in drop down list

↓

Click on view Report.

Query level will be faster than Report level parameter.

How to fix Header:—

Column Groups (Advanced mode) → Static → choose 1st column → F4 (properties of textbox)

→ In properties we need to fix · Fixed data & Repeat on Newpage Should be "true" → ctrl s

→ Export to Excel → If we want the header in all the pages in excel in design tab create header & all header column names & need to enter in header & data remains in same place.

→ Create a shared datasource for a new database (Adventure works)

→ create a datasource for new database (source Adventure)

→ create a dataset for datasource: (set Adventure)

To get all the tables in a database:—

→ Go to query designer.

→ In query designer, select last icon which is having a 't' symbol with table ( ), click on it. it displays

→ It fetches all the tables in a database, add table window, we have to select the table whatever we want.

→ If we select two tables and if that two tables are having primary & foreign key relation, it automatically creates Inner join condition two tables.

Scanned by CamScanner

→ If two tables are not having any relation then it applies crossjoin. b/w two tables.

→ After creating dataset, create a table with all columns & click on preview

→ In SSRS, it doesn't shows nulls, whereever null's are there, it represents empty boxes.

1st ~~sce~~ scenario:-

It should represent 'NA' in empty boxes in SSRS Report.

Ex: start date

Steps:-

1. Select the column which is having empty boxes, Right click on that textbox.

2. Go to textbox properties of that particular textbox.

3. In General, Name & value tables automatically will popup the Selected textbox column (startdate & startdate)

4. Go to Number, we can setup few properties for textbox column in Number.

5. In number category, if we select default option, it will automatically chooses the exact value which is present in textbox column

6. If there is no value in textbox, it will choose NA.

7. In number category except date & time, remaining all are used for decimal & float.

Scanned by CamScanner

How to write a expression to replace empty boxes with 'NA' :-

1. Right click on textbox.
2. Go to expression.
3. In Category, Select the field name which is having empty boxes.
4. write an expression to replace empty boxes with NA

Expression → =iif(isnothing(fields!start.value), "NA", fields!Start.va

5. If ~~column~~ field doesn't have any data, we need to use "isnothing".
6. For isnothing, Go to common functions in category.
7. Go to inspect & Select isnothing (8) If we want in certain format "dd-mm-yy

=iif(isnothing(Fields!start.value), "NA", fields!Start.value)

How to change the background color of column based on reference column values :-

Right click on the field testbox on which we want to change the background color) → directly we cannot write expression for background color.

→ press F4 on textbox. (or) Right click on textbox and select textbox properties.

→ Go to background color & select dropdown option in that click on expression

→ In that expression, we need to select reference column value

→ Based on that reference column values we need to write the expression.

Scanned by CamScanner

Expression: =iif(Fields!organisation level.value=1, "Black", iif(Fields!organisation level.value = 2, "Blue", iif(fields!organisation level.value = 3, "Red", "pink")))

⇒ For selecting the colors, we need to go "constants" in Category

⇒ How to change font color of a textbox on which two rows are having same value:-

→ Press fx on that column, choose font option, color go to expression & select the respective field.

→ To get previous value Go to common-function → Miscellanous → Select previous value.

Expression:-

= iif(Fields! National ID.value = previous(Fields! National ID.value), "Red", "Pink")

⇒ For PDF, to get all columns in one page:-

1. Go to Report & click on report properties

2. change the width & height.

To Remove blank pages:-

→ we need to fit the tablix to the surface of that table without any gap.

How to hide the table which doesn't have data while previewing two tables:-

→ Go to tablix properties.

→ click on visibility option.

→ click on show / hide based on expression,

137

Scanned by CamScanner

=iff(isNothing (Fields ! Business Entity id.value), true, false)

Page break:—

In Excel sheet we have two different table in single sheet, so we need to break sheet & display the tables in different sheets.

↓

→ For this we need to use pagebreak.

→ Right click on tablix → Go to tablix properties

↓

General

↓

Select page break option

↓

checkmark for any of option

↓

click on ok.

How to give sheetname in Excel:—

Click on Fn+F4

↓

click on page name

↓

Provide name for sheet.

How to merge two tablixies in a single sheet:—

→ we need to adjust both the tablix column properties equally in a surface.

→ For sorting the data.

Scanned by CamScanner

Goto tablix properties
↓
Select Sorting
↓
In sortby need to select columnname

Asc (A to Z)

Desc (Z to A)

⇒ How to sort the data when both +ve & -ve signs are there

Ex: Arrange the following in descending order.

1, 2, 3, 4, -5, -6, -7, -8. o/p: -8, -7, -6, -5, 4, 3, 2, 1.

→ Go to tablix properties.

→ click on sorting

→ Add, iif(column<0, -1 * column, column).
↑ True ↑ false

⇒ How to keep decimal points for a value and how to use comma seperator for a value

→ Right click on textbox, Go to textbox properties, choose number and Select decimal places (2) & use 1000 seperator.

→ If we want to show zero as 'NA' then Select Show zero as 'NA'.

→ How to copy code to other field:-

→ Go to textbox properties → Go to format → copy code & then paste it in other textbox

Scanned by CamScanner

⇒ Creating tables:-

→ To get sum of a column:-

→ After creating table, Right click on the header and click on Insert Row, outside group below. & choose:

→ Right click on the row, go to aggregate functions, expression → commonfunctions to get sum of that column, select sum in aggregate functions and select the column.

→ To merge all the cells in a row, select all cells & right click on the cells and select merge cells

→ To split the cells, Right click on the cells and select split cells.

→ If we want to show the sum of two columns in two different rows. we have to add one row.

Grouping:-

→ Creat Report, shared datasource, Create datasource, dataset.

→ Create a table

→ Then add all the columns except the column in which we want to do grouping

→ click on Row groups, Right click on details, click on add group, add parent group, then it displays tablix group.

→ In Group by, select the column in which we want grouping.

→ Add group header & Add group footer.

→ Then grouping column will be automatically added to the table.

→ Double click on grouping column, it will give grouping properties

Scanned by CamScanner

→ check all group properties, if you want to modify anything, modify & click on OK.

→ preview it → Export to excel.

→ To get all parent groups in different pages and to get name on the pages we go for page break.

→ If we have two tables we can go for page breaks, if we have single table then we need to setup that property in group.

→ click ~~on~~ F4 on grouping column → click on group → Select pagebreak

→ Select Breaklocation (between), Select pagename ~~as~~ Right click on page name → select expression → Select field in which we want sheetname

→ pagename should be dynamic.

→ Now preview it, we will get parent groups in different pages and sheet name as orderid & export to excel.

→ To get headers in every sheet, as the grouping column is outside of the table, we need to keep the grouping column inside the table for that we have to add one more row (Inside group - above)

<u>How to add headers for each group in every page:-</u>

→ Add one more row inside the group and copy that header groups in added row & preview it

→ when we create grouping column, we select add header & add footer it will automatically adds ~~one~~ two row, one above the grouping column, one below the grouping column.

→ Below row automatically gives the sum of column values (column = integer)

→ If we want sum of each group column then we have to add row inside the group. If we create row outside of the group it gives total column value but not individual group values.

Scanned by CamScanner

→ How to import the data from Excel to SSRS:— Tasks → Import data

⇒ Drilldown option:— Right click on any column →

→ we have to select least child and add toggle to the parent of that child.

→ Right click on child row and select row visibility and we need to hide the child and keep toggle sign for parent column

Toggle → Display can be toggled by this report item & select the parent

Drill through:—

⇒ we require two rdls for drill~~down~~through.

Tooltip:—

Difference b/w Drilldown Drillthrough

Drilldown	Drillthrough	Subreport
① we need atleast 1 Rdl to do drilldown report.	① we need atleast 2 Rdls to do drill through	① we need atleast 2 Rdls to do subreport
② By using "toggle" item we will do drilldown option.	② By using "action" (or) "Navigation property" we will do drillthrough	② By using "Subreport property" we will do subreport.
③ Incase of drilldown, it will show summary data with a toggle symbol, upon selection of toggle, it will give child details & viceversa.	③ Incase of drillthrough it will navigate from parent to child report. So either we can see parent data (or) child data at a time	③ Incase of subreport we will show both parent & child data in a single page based on filters.

Scanned by CamScanner

How to import data :-

→ Right click on database, go to tasks, click on import, press Next,

→ we need to select import & export.

datasource → import (Microsoft Excel) → path for it.

→ click on next, Destination → Microsoft OLE DB provider for SQL server

→ click on next → Select the tables → click on next → Finish.

Drill through Report:-

→ It is a report that a user opens by clicking a link within another report.

→ we need atleast 2 rdls for drill through report like parent & child rdl.

→ Always we call childreport inside a parent class.

→ create child report & create parent report

Difference b/w Table & Matrix

Table	Matrix
→ In table always we have fixed no. of columns & dynamic rows	→ In matrix, we have dynamic rows & dynamic columns.
	→ Here we take rows as columns.

Report Server:-

Deployement:-

How to generate web service URL (or) Report Server URL

→ Go to start

→ Type Reporting Services configuration manager

→ click on yes.

Scanned by CamScanner

→ Click on connect

→ Go to webservice URL

→ Click on (Webservice URL) apply.

→ Click It will try to connect

→ c

Deployement steps) –

→ copy the report server URL

→ Go to visual studio.

→ Right click on projection in solution explorer

→ Go to properties

→ paste the URL under "target Server URL", provide the target report folder name if it is new otherwise choose the existing folder

→ click on apply and click on ok.

→ If we want to deploy all the reports in a single shot, first build the project then deploy it

→ Right click on project in Solution explorer

→ click on Build

→ once build is succeeded again right click on project. click on deploy

Note:-

→ If we want to deploye single rdl, right click on Rdl & click on deploy

→ After deploying the reports into Report Server URL, and then if we do any changes in the visual studio level, it won't get reflect in the Report Server level until unless we deploy that report.

Scanned by CamScanner

→ Deployement can be done in three ways

1. Bids (Business intelligence developement studio)
2. using upload option in Report Server level.
3. using command prompt.

primary → Parent
foreign → child.

Drill through:-

A drill through report is a report that a user opens by clicking a link within another report.

→ To perform drillthrough, first create child table and then parent table

→ For that parent & child tables create datasource & datasets

→ To filter the data, create report level parameter

How to apply drillthrough option:-

→ Right click on parameter column, go to textbox properties

→ Drillthrough can be done using "Action"./Navigation property.

→ Select "Goto Report" as our child report is RDL.

→ Select "Child RDL" in specify a report category and click on add & Select Parameter (pid) as Name, ~~passing~~ Select the column from the table on which we need to pass the values.

→ We Should Select datatype as integer in Report parameter properties

→ we need to pass something (0) in default values & preview it.

→ when we click on ~~any thing~~ parent column values, that value will be passed to child RDL;

→ In child, we have a parameter, that value will be filtered on top of tablix

Scanned by CamScanner

→ If we want to select any value in parent column, then create we need to go back to design again, instead of that if we create one more Report level parameter for parent table also then we can select directly from child table to go back to the parent table

For websites:-

→ Create datasource and dataset.

→ Go to textbox properties → action → go to URL → expression

iif(Fields!webname.value = "Google", ~~then~~ "https://www.google.com/",

iif(Fields!webname.value = "FB", "https://www.fb.com/",

"https://www.Instagram.com/")

Tooltip option:-

→ Go to textbox properties → Go to general → ~~Expression~~ Effects → Select the column which we want to show when we click another column value

→ Tooltip can be performed only on top of varchar columns

How to get last page data without clicking next button:-

→ we have to create document map, it will keep all the sheet numbers in first page and we have to click on that to get data

→ create document map on top of grouping column, Right click on grouping column → Go to ~~textbox~~ grouping properties

→ Go to advanced and select the grouping column.

Sub Reports:-

→ It is same as drill through, but in drill through we will show either ~~parent~~ parent (or) child.

Scanned by CamScanner

→ Here we can show both (summary & detailed data).

→ create two Rdl's → one subreport (child) & one subreport (parent)

→ create parameter for child & parent both.

→ when we pass a value in parent, it will call child also.

How to create subreport:—

→ Right click on table → Go to subreport → Then wizard appears

→ Right click on wizard → Go to subreport ~~parameter~~ properties

→ Go to general, Select child report (subreport1) and add the parameter also

→ In parameters, we have to select value as parent parameter (but not column ~~values~~ name).

→ Add filters for both parent & child.

→ when we pass any value, it will filter both parent & child.

→ It will show detailed data and summary data in one page.

I Q: How will you pass the parameter from parent to child

A: Right click on the child table, go to subreport properties

→ Go to parameters → we need to all the child's and pass the values from parent.

Cascading Report:—

Cascading is a dependency property, ~~first~~ second parameter depends on first, third depends on first & second.

Ex: Bank example.

→ Create dataset for bankname & parameter (available values)

→ create dataset for states & parameter

→ create dataset for districts & parameter

→ create dataset for branch & parameter

Scanned by CamScanner

FQ: What is forward dependency error

A) we shouldn't select front parameter, we should select back parameter

How to creat matrix:-

→ Right Rows of a table will be taken as columns in matrix. Dynamic rows & dynamic columns will be there in matrix.

→ Go to insert and select matrix

→ If we add any row in physical table, it will be automatically added as column in matrix after refreshing.

Linked Report:- A linked report is a report server item that provides an access to two clients in which one can access all data and other can access only particular data.

Ex: Two clients are there, one client should access all country details and other client should access only one country. In this case we go to Linked report

→ Go to Report, Rig click on eclipse, go to manage, create linked report

→ provide name for linked report → create.

→ In SSRS reports we can see two reports Linked report (chain) will be there

→ change the properties of linked report

→ Go to manage → go to ~~properties~~ parameters & click on use default value & hide it.

Ex: India.

Advantages:-

→ we can hide the data.

→ No need of creating one more report (Rdl) → again & again no need of deploy

Scanned by CamScanner

→ If we delete the parent, then child doesn't work.

Properties:-

Parameters:-

How many parameters are there we can select here. If we want to do any changes we can change here.

→ changes doesn't reflect on visual studio.

Datasource:-

1. Shared datasource 2. custom datasource.

→ when we upload the report ~~manually~~, it will not connect to database we need it will then click on eclipse, select the required database.

when we want to do manually we go to this definitely.

→ If we want to create caching and snapshot records, definitely we need SQL Server authentication. we cannot create these two using windows authentication.

Security:-

→ we cannot give access to everyone. so for security purpose we go to this.

→ Go to Reports, click on eclipse.

→ Go to customise security, click on ok.

→ Add group (or) user.

→ Search on name.

→ If we want to add someone, type name on group (or) user and select the appropriate role.

Sol

→ If we want to change RDL name, we can change RDL name

Scanned by CamScanner

Properties:-

Report timeout:-

use the system default setting:- Generally we lock the system so that no one can access, when we locked the system also, it will be connected to ~~system~~ Server, if they haven't done proper configuration, it will be disconnected to Server. Severdoesn't work.

→ For ex, one report is taking 30 minutes of time, if we ran the report and after 15 minutes if we lock the system and if we use the property as "use the system default setting". While setuping the system they haven't configured, ~~time~~ when we lock our system it should disconnect to server. They made that kind of setting. Then our report gets stopped. 90% we use this option.

2: Allow the report to run for _____ seconds before timing out.

Ex: our report is taking one hour time daily, but one day it took 3-4 hrs and it is loading we have to wait until unless report completes which we cannot wait.

→ our report generally takes one hour and max limit is 1 1/2 hour, within this time it should give report

→ we have to keep (max time in seconds) in this setting so that after even after 1 1/2 hour if the report is loading it says timeout.

→ For this we should know the report ~~time~~ how much time is taking.

Scanned by CamScanner

3. Allow the report to run indefinitely:—

→ For complex reports / we use this long running reports.

When two datasets are having same data, when previewing the data whether it runs parallel (or) serial.

→ It runs parallel. For dependency reports we go to serial.

If we use this "use single transaction when processing the queries" this then dataset runs serially. This will be restricted to this report only. If we use same datasource for other report that also runs serially.

⇒ Go to start → Mstsc → 1. Developement → connect → mostly works on OLTP (online transaction processing)

2. Testing — Less amount → OLTP

3. production — Large amount → OLAP

→ Always select statement should be last in stored procedure.

→ Always atleast one select statement should be there.

Two types of stored procedures:—

1. Reading the data & showing

2. Reading the data & keeping it in temp tables, on top of temp tables we apply logic and we call temp tables.

Caching:—

→ In report server always, it hits the database and fetches the data.

→ Generally in report server level, if we run a report defaultly it will hit the database & get the data everytime.

Scanned by CamScanner

→ There are some reports which will have the changes in a particular time interval other than that timings, the data will be always constant (or) stable. In this cases, instead of going everytime to database we can create cache reports on scheduled basis & get the data from cache.

→ once the scheduled time is over, again it will hit the database for first time and holds the data in cache, also uses the same data from cache till next interval.

History Snapshots :-

→ Generally for long running reports, we will go to snapshot reports. so that we can save the time.

→ create a snapshot report for certain time interval. so that it will keep on create the snapshot with most recent data & hold it in cache

→ when the user wants any report in a particular time then he can just run the snapshots instead of running the report again.

Linked Reports —

→ If one report is using by multiple clients & few clients are ~~working~~ expecting whole data (or) specific data in this time instead of creating multiple reports we can create linked report in server level

Report Servers —

→ we cannot share rdl to everyone. so we have to ~~creat~~ give "URL" to access reports. That is report server.

→ Microsoft has given a link to deploy all reports

Scanned by CamScanner

SSIS

Solution Explorer:-

Project.params:-

By using this option we can create parameters which holds filenames, filepaths, Servername, username & password. These parameters can call in any of the scripts like Csharp, Asp.net, ~~.net etc~~ java, SSIS etc.

These will be with the extension of .params.

Connection Managers:-

→ under connection managers we can create different kind of sources & destination connections. It will be with the extension of .conmgr.

SSIS packages:-

→ under this tab we can create the packages with the extension of .dtsx

Miscellaneous:-

Apart from .params, .conmgr, .dtsx remaining all files will be saved in Miscellaneous.

Control flow:-

→ control flow will check whether workflow is correctly (or) not from one task to another task.

→ It will be controlled by runtime engine in the backend.

→ In controlflow levels we have tasks & containers.

Scanned by CamScanner

Tasks:-

1. Dataflow task.
2. Execute SQL task.
3. Bulk Insert task
4. Execute package task
5. Execute process task.
6. File system task.
7. FTP task. (File transfer protocol)
8. Script task.
9. Send Mail Task.

Control Flow
Data Flow
Parameters
Event Handlers
Package Explorer.

Containers:-

1. For Loop container 2. Foreach Loop container 3. Sequence container.

Dataflow Task:-

→ It will check whether data is flowing correctly (or) not from source to destination.

→ It will be ~~controlle~~ running under dataflow engine in backend & controlled by runtime engine task.

→ In dataflow level we have Sources, transformations, destinations.

Sources:-

1. Excel source 2. flat file Source 3. OLEDB Source

Destinations:-

1. Excel destination 2. flat file destination 3. OLEDB destination.

154

Scanned by CamScanner

Transformations:-

1. Aggregate 2. conditional split 3. Dataconversion 4. Lookup 5. Merge
6. Merge join 7. Multicast 8. OLEDB command 9. Row count 10. SCD (slowing changing dimension)
11. unionall 12. Audit 13. Cache transform 14. copy column
15. Character map 16. Export column 17. Import column 18. Fuzzy grouping
19. Fuzzy lookup

Package Explorer:-

→ In this tab, it will give whole information of package. There will be one more tab gets generate after executing package i-e execution results.

→ Package can be executed using following options.

1. F5
2. start
3. Right click on package & execute package
4. Right click on task & execute task.

Do

Scanned by CamScanner

Datatypes:-

SSIS	SQL
1. String	1. varchar
2. unicode string	2. Nvarchar
3. Four-byte signed Integer	3. Integer
4. currency	4. Money
5. Boolean	5. Bit
6. Double precision	6. Float
7. Two byte signed Integer.	7. Smallint
8. Numeric	8. Decimal
9. Database date	9. Date
10. Database time stamp.	10. Datetime.

Scenarios:-

1. Drop one entire column & execute your package.
2. Rename the existing column & execute the package
3. Add a new column for existing file & run your package
4. In an existing column, change data into diff datatypes & run the package
5. Rename the filename
6. Change the sheetname & execute the package
7. Remove the headers & execute the package.
8. change the database table name & execute the package.
9. Remove the whole content in file except headers & run your package

Scanned by CamScanner

10. Remove entire headers & data & run the package.

11. Keep the diff datatypes b/w source & destination

12. change the datatypes in dataconversion & run the package.

Scanned by CamScanner

SQL

SQL - Structured query language.

SQL is a domain specific language and data base tool which is used to create and access database to support sw application.

Database → collection of data
Systematic

DBMS → collection of programs which enables its users to access database manipulate data, help in representation of data.

Ex: FB, Instagram, telephone directory, Electricity service provider.

first → 1960 → Charles Bachmen's → first DBMS is IDS
↓
Integrated data store

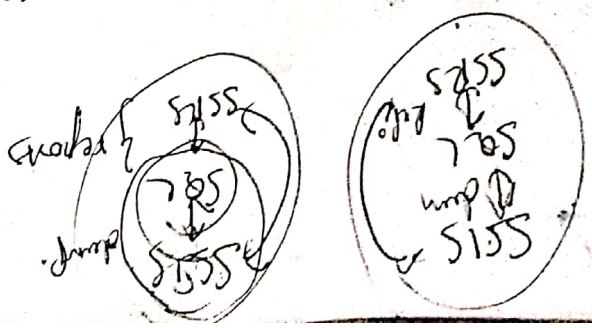
MSBI – Microsoft business Intelligence.
↓
which provides best solutions for Business intelligence & data mining queries.

MSBI →
- SSIS → SQL Server Integration Services
- SSAS → Analysis
- SSRS → Reporting tool.

Normalisation

(3)

⇒ we shouldnot store any tables in 'tempd' in system databases becoz if SQL gets restarts



156

Scanned by CamScanner

System database:-

⇒ All system related objects which gets stored in master database.

⇒ ~~we shouldn't store any tables~~

Model data base:-

→ Default model values for all the objects in SQL Server will comes from model data base

MSDB = Microsoft database → objects related to SQL Server jobagent
↓
SQL Server Agent

& few other things will get stored in MSDB.
↓
related to back up & restore information

tempdB = To store all temperory table.

→ It is faster becoz it won't write Log file in database
↓
Becoz it is recreating

```
If exists (select top1 from deleted)
begin
set @action='record is deleted'
end.
```

⇒ why don't we use schemabinding in other views.

⇒ Physical-CI (PIC), Indexview

⇒ ~~x~~y don't we use ~~x~~ in indexview.

IV ≥ minV & < maxV.

~~Create~~ Sck

Create sequence Sequencename
Start with initial value
increment by increment value
minvalue minimum value
maxvalue maximum value
cycle/ Nocycle;

160

Scanned by CamScanner

Apollo Hospital Branches (1)

Br NO	Br Address
1.	2nd street, Kakatinagar, Hyd
2.	Global village Banglore
3.	Arundalpet 7th layout delhi
4.	9th street prakashNagar Mumbai
5.	Karapakkam chennai

(PK) (P1) (2)

B.NO	Br. manager ID	Manager Name	Br contact NO
1.	1.		
2.	2.		
3.	3.		
4.	4.		
5	5.		

Doctor details

Br. NO	DOCID	DName
1.	1	ThomasRonald
1.	2	David Kelvin
2	3	Kerin Mronald
4	4	Nelson cristi
5	5	Nithin patekar
6. (F1)	6 (P2)	Ashareddy

(5)

Doc ID	Docspecialisation	Doc Avability
1.	Asthmaspecialist	Everyday
2.	Dermatalogist	Mon, Tue, Wed
3	Anesthasit	Tues, Thurs, Saturday.

(F5)

Patient details

Br NO	PID	PName	PADD	PEmail	pmob
1.	1.	Vimal	Bradipet	Vimal@gmail	9100536
1	2	Mukesh	Rangareddy	Mukesh@gmail.com	826745
2	3	Komal	Marathahalli	Komal@gmail.com	976250
3	4	Nayan	Sholinagur	Nayan@gmail.com	12345
5 (P2)	5 (P3)	Kunal	PK Nagar	Kunal@gmail.com	987642

(3)

PID	P. Disease	IP	OP
1.	Asthma	yes	NO
1.	Eyesight	NO	yes
2.	Diabetes	yes	NO
3.	Hearing & ear	NO	yes
4. (P3)	Heart	yes	NO

(4)

PID	Poperation	PBill	P.P.A	PDue
1.	Asthmaopeciation	10000	5000	5000
1.	Lazor	50000	50000	0
3.	Eargera	10000	10000	0
(F4)				

Scanned by CamScanner

system defined storeprocedures:

1. Sp- Rename.

2. scope- Identity - To find last inserted value in identity table.

Scanned by CamScanner

System defined functions:—

1. GetDate() → upto milliseconds
2. SysDatetime() → upto nanoseconds.

Scanned by CamScanner

Global variables:-

@@Identity - It shows last inserted value in identity table.

Error messages:-

1. Truncation binary error → If varchar size exceeds beyond the limit then it will throw this error.

2. Arithematic overflow → If we are giving integer numbers more than the limit in the case of integers, float, money and decimal.

3. Dataconversion error:- conversion failed

4. Database could not be exclusively locked to perform the operation.
↓
unable to change database and time.

5. changing anypart of an object name could break scripts and store procedures. → changing table name.

6. checking identity information, current identity value 'NULL'
DBCC execution completed. If DBCC printed error messages, contact your system - administrator.

Scanned by CamScanner

1. cannot insert explicit value for identity column in table 'tablename' when identity insert is set to off.

Scanned by CamScanner

Indexes:- Indexes helps us in reaching data faster. There are two types of Indexes.

1. Clustered Index
2. Non-Clustered Index.

Heap:- A table without having a clustered index & non clustered Index.

Clustered Index syntax:-

Create clustered Index < Ix.Name>

ON Tablename(columnName)

Non-clustered Index:-

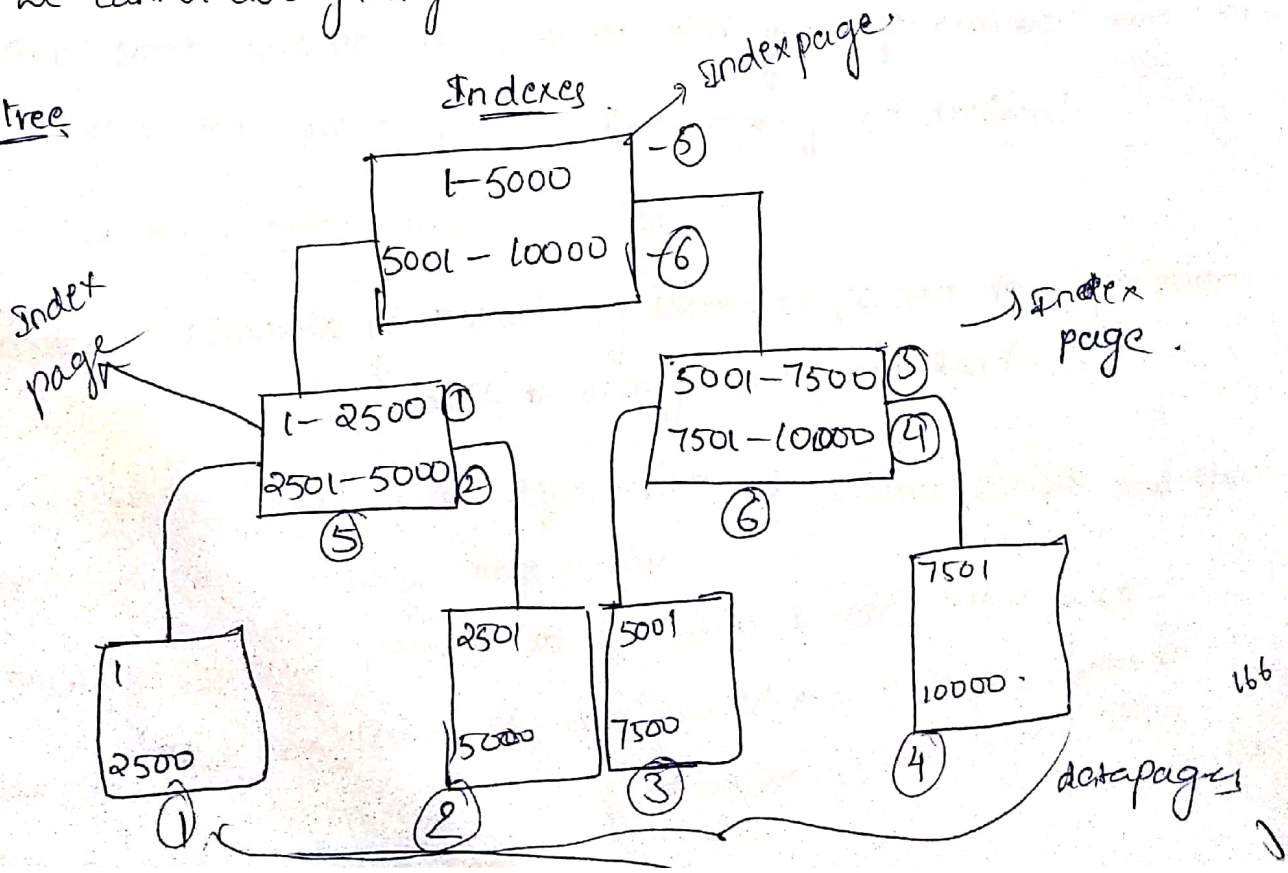
Create Non Clustered Index < Ix.Name>

on Tablename(columnname)

⇒ We cannot update/remove/modify/rename for Indexes

→ we cannot do anything in indexes

B+tree



Scanned by CamScanner

⇒ Sca

Navi

→ Every non clustered Index, there will be one key clustered

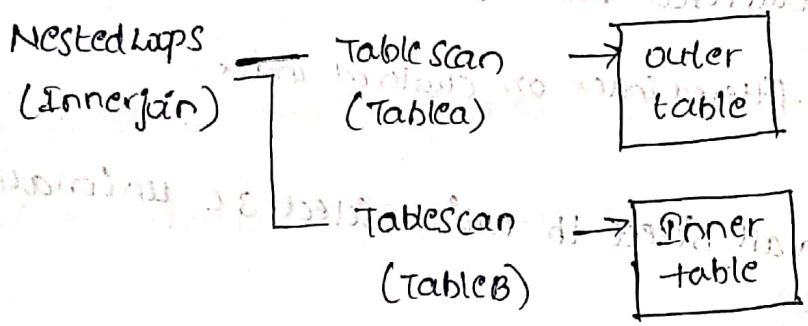
Nested Loop joins:-

→ compares each row from outer table to each row from the inner table and looks for rows that satisfy the join predicate

→ usually, SQL Server Selects smallest i/p as outer table

→ No memory required

→ Cannot scan well.



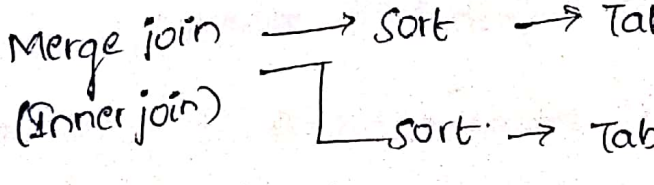
Merge join:-

→ Needs atleast one Equality predicate

→ Needs data to be ~~stored~~ sorted in any order or join predicate

→ May require memore for sorting

→ May use tempdb for sorting if memory is not sufficient.



167

12

Scanned by CamScanner

Hash join:-

→ Needs atleast one Equality predicate

→ Hash join scales (usually) good in datawarehousing scenarios handle Large amount of data that SQL cannot

→ Needs Memory to store in memory hashtable (Build Table)

→ Hashjoin is a blocking operator, cannot process rows until Build table is ready.

Hashmatch (Innerjoin)
- Tablescan (Table A)
- Tablescan (Table B).

⇒ A data without clustered index is called Heap.

⇒ we cannot create filtered index on clustered index.

⇒ Select 1A union all Select 1b union Select 3 C unionall

Select 1A except Select 5b o/p: 1, 3.

4
1
3

168

3

Scanned by CamScanner

```
declare @ date varchar(100) = '1/1/28'
select case when len(substring(@date,1, charindex('/',@date)-1))=1 then '0'
+substring(@date, 1, charindex('/',@date)-1))
else
substring(@date, 1, charindex('/',@date)-1) end + '/' +

case when len(substring(@date, charindex('/',@date)+1, (charindex('/',@date,
charindex('/',@date)+1) - charindex('/',@date))-1))= 1 then '0' +
substring(@date,charindex('/',@date)+1, charindex('/',@date, charindex
('/',@date)+1 - charindex('/',@date))-1
else
substring(@date, charindex('/',@date)+1, charindex('/',@date, charindex
('/',@date)+1 - charindex('/',@date))-1 + '/' +
                                charindex('/',@date)
case when len(substring(@date, charindex('/',@date)+1)+1, charindex
('/',@date, charindex('/',@date)+1)+1 = 2 then '20' + substring
(@date, charindex('/',@date, charindex('/',@date)+1)+1, charindex
('/',@date, charindex('/',@date)+1)+1
else
~~charind~~ substring(@date, charindex('/',@date, charindex('/',@date)
+1)+1, charindex('/',@date, charindex('/',@date)+1)+1)
end.
```

169

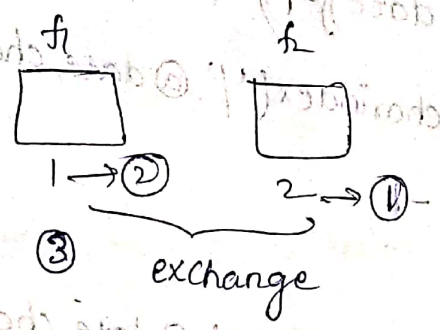
Scanned by CamScanner

Statics → No. of records in table (when we create Index automatically statistics will be created in backend

⇒ SQL Server doesn't know estimated no. of rows in table variable, so it gives 1Mb to tV → wrong estimation.

⇒ It maintains statistics, it knows estimated no. of rows so for larger data we go for templables.

Case:-



→ we can avoid nested loop join using covered index.

outer query } for each row.

Inner query

tempdb → Hash match.

tablevariable → Nested loop.
(Smallest table as outer table)

Scanned by CamScanner

Backup & Restore:-

1. Full 2. Differential 3. Log.

Full backup → Entire copy of database backup is called full backup. (once in a week)

Differential :- It contains data since last full backup.

Log :- It copies of transaction log.

Syntax for database backup:-

Backup database databasename to disk = 'path/DBname_Full_Date.bak'

Differential database backup:-

Backup database databasename ~~to disk~~ = 'path/DBname_Diff_Date.bak'
with differential

⇒ MSDB will maintain when backup is done

Log database backup:-

Backup Log databasename to disk = "path/DBname_log_dat.Trf'
↓
It is better

Copy only database:- It doesn't interrupt the differential backup.

Syntax to restore:-

Restore database databasename.

Scanned by CamScanner

19/1/2017

Transactions

Transactions are two types 1. Implicit (auto committed)
2. Explicit.

Concurrency problems:- whenever multiple users want to access the Same data in a table then concurrency problems occurs. To avoid that we go to Isolation.

Trans1	Trans2
Begin Trans	Select * from T1 where ID=1
update T1 Set (Ramesh) ename = vinod where ID=1	o/p → Name = vinod
Rollback.	Dirty Read ↓ which gives wrong information.
o/p → Name = Ramesh.	

Committ → Make it permanent

Rollback → Go back to the previous stage where before transaction started.

Isolation levels:-

1. Read uncommitted
2. Read committed
3. Repeatable Read
4. Serializable

(when one transaction is working other transaction is not allowed) Pessimistic Concurrency Model

default ↓ Read committed.

5. Snapshot → optimistic concurrency model. → don't want any problems with high concurrency.

172

Scanned by CamScanner

ACID Property:— Set of properties

These three will be automatically performed by SQL

A → Atomicity → do it complete (or) don't do anything
Don't leave the transaction in between.

C → consistency → shouldn't leave the database in consistency state

I → Isolation → Each & Every transaction should be isolated with eachother

D → Durability → data should be there after system failures due to logfiles.

This is we have to perform.

Isolation:—

Read committed →

T_1	T_2
Begin Tran	Select * from T. where ID=1
(paarth) Set Name = 'Madhu' where ID=1	o/p → waiting state

Which gives committed data after updating only it will give o/p

How to change the database transaction level:—

Set Transaction Isolation level Read uncommitted

Select * from T_1 where ID=1

Dirty Read → Single transaction

Non Repeatable → Multiple transactions

Trans1	Trans2
Begin Trans	
Select Sum(sal) as totalsal from emp 100000	Begin Trans
	update emp
	Set Salary = Salary + 10000
	commit
Select Sum(sal) as totalsal from emp. 200000.	

Non Repeatable Read → To avoid this do Trans1 under Repeatable Read.

Scanned by CamScanner

Trans1	Trans2
Begin Trans1	(2) Insert into T_1 (1), (2)
Delete from T_1 where (1) ID in b/w 1 & 10	Inserted
Select * from T_1 where (3) ID in b/w 1 & 10.	

O/P: (1) (2)

Phantom Reads. Trans1

To avoid this, Run under ~~Trans1~~ Serializable

X → Exclusive

U → update

S → shared.

Read uncommitted → Highest concurrency

Serializable → Lowest Concurrency.

	DirtyRead	Nonrepetable	PhantomReads
Read uncommitted	✓	✓	✓
Read committed	X	✓	✓
Repeatable Read	X	X	✓
~~Non Repeatable Read~~			
Serialization	X	X	X
Snapshot	X	X	X

Trans1	Trans2
Begin trans	Delete
update emp	

Scanned by CamScanner

Trans1 Trans2

Trans1 need Tran 2

Tran 2 need Tran1

Both will not get completed

this is called deadlock.

SQLMaestros

SQLServerGeeks

9.0032853-
9.0067338

SQL Server Selects the transaction, which can be rollback easily as deadlock victim. (we can avoid it by using proper indexes)

Transaction → Every query batch that runs in a SQL Server

1. Implicit — Auto committed (query is committed)

2. Explicit — Starts by mentioning Begin Tran and ends by committ / Rollback.

ACID properties:—

Atomicity → It must be atomic unit of work. Do everything (or) don't do anything.

Consistency → whenever crash happens, it shouldn't leave there, it should go back to before transaction state.

Isolation:— A transaction should see the data only before or after the concurrent transaction gets completed, not when transaction is in progress.

Durability:— All the transactions should be persisted despite system failure

Lock hierarchy:—

1. Database level locks
2. Table level locks
3. Page level locks
4. Row level locks.

Intent exclusive lock.

175

Scanned by CamScanner