SQL

120CS0121
B.N.Pravallika

Databox Creat^n

- Command based · • GUI Based

Create Database Namedb

Use dbName

→ Create Table

Create Table Name
(Attribute1 Datatype,
" 2 " ,
---)

→ Insert

insert into table_Name values
(value1, values2, --)

→ select, Delete

select * From Table_name

Delete From Table_Name where cond^{n}1, cond^{n}2, ---

DDL:

used to defined or alter the structure of db items

Create, ALTER, DROP, RENAME, TRUNCATE, COMMENT

comes under DDL (Data definitn Language)

DML: (manipulatn)

for accessing & manipulating data in db.

→ select, Insert, Update, Delete, MERGE — comes under DML

* DROP Table Table_Name

* **Where:**

to specify cond^n while fetching data from single table or by Joining with multiple tables

select column1, --, columnN from Table_name where [cond^n];

Operators

AND:

Select Column1, ...Column N From Table_Name where [Condn] AND [Cond 2] AND ... AND [Condn];

OR, ...

DELETE:

Delete from Table_Name where [Condn];

ALTER:

AL allows us to add, modify, delete columns from existing table

ALTER Table Table_name DROP Column column_name;

NOT NULL:

Constraint enforces a Column to NOT accept NULL values

Create Table Name (Column1 datatype NOT NULL, ...);

CHECK:

To check whether value entered into a record.

Create Table

Table_name [Column1 datatype CHECK [condn], Column2 datatype, ... , column N datatype Contraint);

Unique:

ensures that there are no duplicate Values present in Column.

Create Table Table_name (Column1 datatype Unique, Column 2 datatype, ...

1° key:

uniquely identifies each row in a table

NOT NULL	UNIQUE	1° key
more then 1 values Can assign as not null	>1 unique key	only 1
duplicates ✓	Can accepts NULL value	Can accept Null

ALTER:

To add, delete/drop, modify column.

Alter - ADD:

ALTER Table Table_name ADD Column name_1 data type, 2 datatype, ---;

ALTER-Drop:

Alter table table_name

Drop Column column_name;

-Modify:

ALTER Table table_name

Alter Column column_name datatype;

TRUNCATE:

used to mark the extents of a bable for deallocation (Empty for reuse).

TRUNCATE Table table_name;

RENAME:

EXEC SP_rename 'old name', 'New name';   (Table)

EXEC SP_rename 'Tablename.Old Columnname', 'NewColumn name', 'Column';   (Column)

Comments:

Single line: -- Single line

multiline: /* multiline */

Inline: select * from /* customers; */

DCL (control)

used to grant & take back authority from any database user.

GRANT, REVOKE;

Grant:

Grant privileges_names ON table_name TO user;

REVOKE:

REVOKE privileges_names ON Table_name FROM USER;

DQL : (Query)

for performing queries on the data within schema objects

- include SELECT commands

* select column1, column2 from table name;

select * from table name;

Data types.

Value represent : INT, CHAR, DATE, VARCHAR

Space & memory occupy : Fixed & variable

Datatype → Numerical data type

→ String data type

→ Date & Time datatype

Funcn for Date & Time data type:

1. DATE DIFF
2. GET DATE
3. GETUTCDATE : Return the current UTC date & Time
4. SYS DATE TIME: Return date & time of SQL server
5. DAY : return day of month for date
6. MONTH : return month part of date
7. YEAR : return year part of date.

Eg. DateDiff (interval, 'date1', 'date2')

Getdate ()

Day (date)

To Calculate ·age from DOB:

UPDate Table-name
SET year_differen =
Date diff (year , dOB , GETDATE())
where doB IN (SELECT· doB FRom Table name)
SELECT CONVERT (varchar(10), CONVERT (DATE, dob , 105), 105)
FRom DOB

SQL operators

Aggregate Func^n

Perform a calculatn on a set of values as returns a single value.

Approx_Count_Distinct AVG VAR COUNT_BIG
CHECKSUM_AGG COUNT MIN GROUPING
GROUPING_ID SUM MAX STDEV.

Approx-count- distinct()

returns approximate no. of unique non-null values in a grp

Select ____ (column name) AS Alias name FRom Table_Name ;

Count:

to count no. of rows in dB

Select COUNT (*) FRom table_name;
select COUNT (column name) From tableName;
select Count(*) From TB where [condn];
select Count (DISTINCT · column name) FRom Table_Name;
select Count_BIG (*) FRom TB;

Sum;

select sum (colum_name) FRom TB ;
select sum (column_nar) From TB where (condn);

AVG():

select AVG (column name) From TB;

MIN():

select MIN (column_name) From TB;

MAX:

↗

CHECKSum_AGG():

select checksum_AGG (column_name) From TB;

Grouping:

A specifred column expression in a Group by list is aggregated or not.

Grouping returns 1 for aggregates

0 for non aggre in result set

select Grouping (column name) From TB group by column_name;

Grouping_ID():

concatenates the o/p of Grouping Funcn applied to all colum

STDEV():

returns statistical standard deviation of all values in specified expression

select STDEV (column_name) From TB;

VAR():

returns variance of all values

String manipulation Funcn,

It is a built in funcns which take an i/p string & returns a string or numeric value.

* ASCII(): returns ASCII code value of leftmost characte
* CHAR(): returns single byte character with specified integer code.
* ~~Char Index~~():

select ASCII (column name) As Alias_name FROM TB;
CHAR

* CHAR INDEX( ):

searches for 1 character expressic inside a second character - -

* PATINDEX;

return start positn of first occuran of a pattern in a specified expressn or zeros if the pattern is not found.

select column_name Charinden (exp to find, exp to secure) From TB;
Patinden

* CONCAT( ):

Joinin of 2/more string values in an end-to end manner

* · Select CONCAT (Column name 1, ' ', Column_name 2) As Alias_name
From TB;

* DIFFERENC( ):

* LEFT( ):

returns the left part of character string with specified no of characters.

select LEFT (column name, 3) AS Alias_name From TB;
right
UPPER
LOWER → return char exp after convertn upper case to lower case
LEN

LTRIM( ):

return a character exp after it removes leading blanks

RTRIM( )

" " " " truncate all trailing spaces

select LTRIM (Column_name) From TB;
RTRIM

TRIM( )

remove space character char(32) or other specified characters from start & end of string

select TRIM (Coolumn_name) As Alias_name From TB;

SET operators

Union: , union all , Intersect .

select (Column_name1 , 2 , --) From Table1.
UNION
(column_name 1 , 2 , ---) From Table 2.;

EXCEPT:

returns any distinct values from left select query that are not a found on right select query

select Column1 , Column 2 From Table-1
EXCEPT
select Column1 , " " " -2

RIGHT-set operator

IN , BETWEEN, LIKE , EXISTS , IS NULL, SOME/ALL/ANY

IN: search to specified value matches any value in set of multiple values

select column_name(s) From TB where column name IN (select statement);

Between:

used to get values with in a range

select * From tB where Column -name between value1 And value2;

LIKE:)

SQL Constraints

NOT NULL, Unique, --- 10 Key, --

SUB Query:

used to return data that will be used in main query as a condn to further restrict data to be retrieved

```
select column_name(s)
From TB(S)
Where column_name operator ⟶ =, <, >, >=, <=, IN, NOT etc
(select column_name(s)
 From table_name(s)
 [where condn]
   )
```

WITH:

WITH Clause allow you to give a sub-query block a name which can be referenced in several places within the main SQL query.

ALIASES (AS)

are used to give a table, or a column in other table, a temp name.

DISTINCT:

keyword used to eliminate all duplicate records & fetching only unique records

```
select DISTINCT column_names(s) From TB;
```

Group By

→ Used to arrang identical data into grps with the help of some funcns.

→ used usually for purpose of performing 1/more aggregatns on each grp.

Select CN(S) from TB GroupBy column name(s)

Group by Rollup:

In addn to group by, it "rolls up" results into subtotals grand totals.

- For group by rollup (a,b) the results have groups for unique values of (a,b), (a,NULL); & (NULL, NULL);

Select Column name(s) From TB Group by Roll up (NCS).

Group by CUBE:

Creates groups for all possible combinatns of columns

For gr, (a,b) (NULL, b), (a, NULL), & (NULL, NULL);

SELECT CN(S) From TB Group By CUBE (CN(S))

HAVING

specifies search condn for a grp of aggreate.

Order by.

→ Sorts data returned by query in SQL server.

→ By default, ORDER by sorts the data in ascending order

→ we can used keyword DESC to sort the data in descending order & keyword ASC to sort in ascend

SELECT CN(S) From TB ORDER BY CN ASC/DESC

WHERE

used to specify a condn while fetching data from a single table / by Join's with multiple tables

SELECT colm-name (s) From TB Where Condn

JOINS

JOINS, helps us to Combine 2/more tables to perform various operatns

→ without SQL Joins, there is no concept of data normalization b/c we have to store the entire data in 1 table.

6 types

1) **INNER JOIN:**

returns rows present in both the Left table & right table only if there is a match. otherwise zero.

SELECT CN(S), From TB INNER JOIN TB ON Condn

2) **LEFT (OUTER) JOIN:**

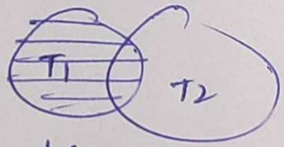
returns all rows in LEFT table & matching rows from Right.

SELECT CN (S) From TB LEFT JOIN TB on Condn

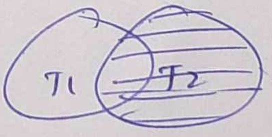
3) **RIGHT (OUTER) JOIN:**

Same as above. _ ~

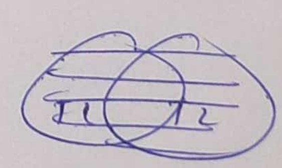
So



LEFT JOIN



RIGHT JOIN



FULL OUTER

4) **FULL OUTER JOIN:**

return result contains rows from LEFT & RIGHT ,

SELECT select_list from T1 FULL OUTER JOIN Table 2 ON JOIN_predicate.

5) **Self join:**

allows us to join a table to itself.

SELECT select_list from T1 INNER JOIN T2 on join_pre

6) **Cross join:**

produces cartesian product of 2/ more tables

SELECT select_list from T1 CROSS JOIN T2 ;

Update join:

Join clause along with update stateme

INNER JOIN / left outer join is used in update JOIN

Delete join:

INNER JOIN / left outer join used in delete join

DELETE table_A from table_A [INNER | LEFT] JOIN Table_B ON Predicate where predicate ;