SQL(STRUCTURED QUREY LANGUAGE)

**CHAPTER 1**

* DDL:
* DML:
* DCL:

1. **DDL**(DATA DEFINITION LANGUAGE):
   1. Create
   2. Alter
   3. Drop
2. **Create: ( create the table)**

Ex: create table csc2(id int, course char(10), fees int);

select \* from csc2

1. **Alter: ( add the table column)**

ex:alter table csc2 add subject char(10);

**View: ( view the table)**

select \* from csc2

1. **Drop: ( delete the table)**

Syntax: drop table tablename

ex:drop table csc

1. **DML**(data maniPulation language):
2. Insert
3. Select
4. Update
5. Delete
6. **Insert:**

Ex:insert into csc2 values(567,'HDCA',8400,'MSOFFICE')

**2.SELECT:**

**i) ALL:**

select \* from csc2

**ii) VIEW THE COLUMN:**

select course,fees from csc2

**iii)CHANGE THE COLUMN NAME:**

select id AS studentid from csc2

**iv)TOP:**

SELECT TOP 2 \* FROM EMP

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 343 | Manager | 50800.00 | NULL | sasa | 12345 |
| 454 | Clerk | 5000.00 | 45.0000 | NULL | NULL |

**v)WHERE**

**1)(=) Equal to**

select \* from csc2 where course='ADJP'

**2) (>) Greater than**

select \* from csc2 where fees>6000

**3) (>=) Greater than or equal to**

select \* from csc2 where fees>=4000

**4)(<>)**

select \* from csc2 where fees<>5230

**vi)LOGICAL OPRATORS**

**1)OR:**

select \* from csc2 where course='ADJP'or course='HCA'

**2)AND**

select \* from csc2 where course='ADJP'and course='HCA'

**3)BETWEEN**

select \* from csc2 where fees between 7000 and 10000

**4)NULL:**

select id, fees from csc2 where subject is null

**5)NOT NULL:**

select id, fees from csc2 where subject is not null

**6)IN:**

select \* from csc2 where fees in (5230,8400)

**7)LIKE:**

**a)**select \* from csc2 where course like 'd%'

Result:

|  |  |  |  |
| --- | --- | --- | --- |
| 342 | DCA | 6900 | TALLY |
| 453 | DMO | 4546 | EXCEL |
| 123 | DTP | 4790 | PHOTOSHOP |

**b**)select \* from emp where empname like 'r[e-s]%'

|  |
| --- |
| 565 md 20000.00 56.0000 reema 12347 |
| 788 teacher15000.00 12.0000 reka 12348 |
| 888 doctor5000.00 34.0000 reshma12349 |
|  |

**3.UPDATE:**

**i) change the value:**

update csc2 set course='h&n', subject='hardware' where fees=5324

**ii)CALCULATE THE VALUE:**

**create database honey ----------- >database creation**

use honey --------------- **> use the database**

**table creation:**

create table san(sno int, id int, name char(20), mark1 int, mark2 int, mark3 int)

**insert the table values:**

insert into san values(1,1001,'sasi',98,97,87)

insert into san values(2,1002,'sasika',98,97,88)

insert into san values(3,1003,'josphine',97,94,84)

insert into san values(4,1004,'devil',34,35,64,null,null,'')

**select the table:**

select \*from san

**alter the table:**

alter table san add total int

alter table san add average int

alter table san add result char(10)

alter table menaka add rank int

**update the table:**

**total calculation:**

update san set total= mark1+mark2+mark3

**average calculation:**

update san set average= total/3

**result calculation:**

update san set result= 'pass' where mark1>=35 and mark2>=35 and mark3>=35

update san set result= 'fail' where mark1<35 or mark2<35 or mark3<35

**rank calculation:**

update menaka set rank=1 where result='pass'and total>=230

update menaka set rank=2 where result='pass'and total>=220 and total<230

update menaka set rank=3 where result='pass'and total>=200 and total<210

update menaka set rank=0 where result='fail'

1 1001 sasi 98 97 87 282 94 pass

2 1002 sasika 98 97 88 283 94 pass

3 1003 josphine 97 94 84 275 91 pass

4 1004 devil 34 35 64 133 44 fail

**iii) incrementing the value:**

update csc2 set fees= fees+500 where course='dmo'

**iv) CHange the full column values:**

update csc2 set fees= fees+500

**v)assigning the null value:**

update csc2 set course=null, subject='hardware' where fees=5824

**vi)ASCENDING ORDER**

select \* from csc2 order by fees asc;

**vii)decesding order**

select \* from csc2 order by fees desc;

**viii)copy to another new table**

select empname, salary, description into new from emp

**4.Delete:**

1.delete adds from csc2------ > column delete

2.delete id=111 from csc2------- > row delete

3.delete table csc2 ---- > table delete

**DCL(data control language)**

1. Grant
2. Revoke

**i)grant:**

grant update on csc2 to public

**ii)revoke:**

revoke update on csc2 to public

**Chapter 2:**

**t(transact)-SQL FUNTIONS:**

1.SINGLE row (or) scalar FUNCTIONS:

2.AGGREGATE (or) group FUNCTIONS:

**1.Single Row Functions:**

1. Numeric Functions;
2. String Functions
3. Date And Time Functions
4. Calculating Results
5. Conversion Funtions

**1.numeric functions:**

ex:

* select pi( ) (ans= 3.1415926535897931)
* select sin(90) (ans = 0.89399666360055785)
* select ceiling(12.34) (ans= 13)
* SELECT floor(12.34) (ans= 12)

**2.String functions**:

ex:select upper(name), upper(author) from book

ans:

|  |  |
| --- | --- |
| SQL | JOHN |
| TALLY | JOE |
| C++ | BALA |

* select left('quries',2),right('quries',2)

ans:

qu

es

* select len('quries') ans: 6

**3.date & time functions:**

ex:select getdate() 'todays date'

ans: 2005-01-01 01:14:22.187

select getdate()+ 5 '5 days from today'

ans: 2005-01-06 01:22:45.090

**4.calculating functions:**

create the table and select the calculations

ex: student table

select name , m1+m2+m3 ‘total’ from student

ans:

|  |  |
| --- | --- |
| name | total |
| abi  bala | 456  345 |

**5.conversion functions:**

select \* from student

|  |  |  |
| --- | --- | --- |
| code | Name | city |
| 232 | abi | pmk |
| 675 | anu | rmd |

select name+'code is '+convert(varchar,code)from student

ans:

|  |  |
| --- | --- |
| abi | code is 232 |
| anu | code is 675 |

**ii) aggregate functions(or) group functions:**

1. sum
2. avg
3. count
4. count(\*)
5. max
6. min

ex:select \* from csc2

id course fees subject

|  |  |  |  |
| --- | --- | --- | --- |
| 342 | DCA | 6900 | TALLY |
| 567 | HDCA | 8900 | MSOFFICE |
| 453 | DMO | 4546 | EXCEL |
| 897 | ADJP | 5730 | JAVA |
| 123 | DTP | 4790 | PHOTOSHOP |
| 234 | NULL | 5824 | HARDWARE |
| 867 | HDUA | 6780 | SQL |
| 567 | HDUA | 8400 | C |

**sum,avg,count(\*) :**

select sum(fees) ‘sum of fees’,avg(fees), count(\*) from csc2

|  |  |  |
| --- | --- | --- |
| no column name | no column name | no column name |
| 51870 | 6483 | 8 |

**count;**

SELECT COUNT(fees) FROM CSC2 ANS: 8

**max, min:**

select max(fees), min(fees) from csc2

|  |  |
| --- | --- |
| 8900 | 4546 |

**2.Group by:**

select max(fees), min(fees) from csc2 group by course

|  |  |  |
| --- | --- | --- |
| 5824 | 5824 | Null |
| 5730 | 5730 | adjp |
| 6900 | 6900 | dca |
| 4546 | 4546 | dmo |
| 4790 | 4790 | dtp |
| 8900 | 8900 | Hdca |
| 8400 | 6780 | hdua |

**compute:**

select id,course,fees from csc2 order by subject compute sum(fees)

\

|  |  |  |
| --- | --- | --- |
| Id | course | fees |
| 567 | HDUA | 8400 |
| 453 | DMO | 4546 |
| 234 | NULL | 5824 |
| 897 | ADJP | 5730 |
| 567 | HDCA | 8900 |
| 123 | DTP | 4790 |
| 867 | HDUA | 6780 |
| 342 | DCA | 6900 |

sum 51870

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Empid | description | salary | unitprice | empname | deptno |
| 343 | manager | 50800.00 | NULL | sasa | 12345 |
| 454 | clerk | 2000.00 | 45.0000 | NULL | NULL |
| 565 | md | 20000.00 | 56.0000 | reema | 12347 |
| 788 | teacher | 15000.00 | 12.0000 | Reka | 12348 |
| 888 | doctor | 95000.00 | 34.0000 | reshma | 12349 |

**COMPUTE BY:**

select empid,description,salary from emp order by deptno compute sum(salary)by deptno

output:

Empid description salary

454 clerk 45000.00

sum

45000.00

Empid description salary

343 manager 50800.00

sum

50800.00

Empid description salary

565 md 20000.00

sum

20000.00

Empid description salary

788 teacher 15000.00

sum

15000.00

**Chapter 3:**

**ADvanced query concepts**

**1.union** (The Common Column Name Only Accepted)

ex:select empname from emp union select name from student

NULL

abi

anu

reema

reka

reshma

sasa

**2. union all**(Two Table Values Allowed)

ex:SELECT EMPNAME,empid FROM EMP UNION all SELECT NAME,code FROM STUDENT

|  |  |
| --- | --- |
| NULL | 454 |
| reema | 565 |
| reka | 788 |
| reshma | 888 |
| abi | 232 |
| anu | 675 |

**3.joins:**

**i)inner join**

select \* from book inner join computer on book.name= computer.name

**2.OUTER JION:**

1.left outer join

select b.name,size,rate from computer c left outer join book b on c.name=b.name

**2.right outer join**

select b.name,size,rate from computer c right outer join book b on c.name=b.name

**3.full outer join**

select b.name ,size ,rate from computer c full outer join book b on c.name=b.name

**sub qureies:**

select \* from book;

select \* from computer;

**1.SINGLE**

ex:1select \* from book where price=(select min(price) from book)

Ex:2select \* from emp where salary=(select salary from emp where empid=565)

565 md 20000.00 56.0000 reema 12347

**in:**

ex:1select \* from emp where salary in(select salary from emp where empid=565)

ex:2select \* from book where name in (select name from book where price=630)

**not in:**

select \* from emp where salary not in(select salary from emp where empid=565)

**ALL(ANY VALUE U WILL GIVEN)**

select \* from book where price>all(select price from book where price=400)

**ANY(GIVEN VALUE IS MUST BE ENTEREDIN THE TABLE)**

select \* from book where price>any(select price from book where price=550)

**CHAPTER 4**

**DATA INTEGRITY**

CORRECTNESS OF DATA STORED IN THE DATA BASE.

FOUR TYPES:

1.Entity Integrity --> Primary Key Constraint,Unique Constraint

2.Domain Integrity --> Default Constraint, Check Constraint

3.Referencial Integrity-->Foreign Key Constraint

4.User-Defined Integrity-->Rules, Stored Procedures

**CONSTRAINTS:**

It Is A Rule Deined For A Single Table That Limits The Possible Values User Can Enter Into The Table Or Column.

FIVE TYPES:

1. PRIMARY KEY CONSTRAINT
2. UNIQUE CONSTRAINT
3. FOREIGN KEY CONSTRAINT
4. CHECK CONSTRAINT
5. DEFAULT CONSTRAINT
6. **PRIMARY KEY CONSTRAINT:**

noTES: i) null values not allowed

ii)only one primary key defined per table

create table csccourse(cid int constraint pk\_cid primary key, cname char(10),description char(20),fees int)

insert into csccourse values(676,'dca','tally',4232)

545 ADJP JAVA 6500

676 DCA TALLY 4232

**2.UNIQUE CONSTRAINT:**

Notes: Null Value Allowed But Only One null value per Row

create table shop(shopid int,sname char(10) constraint uk\_sname unique,city char(20))

insert into shop values(NULL,'HDCA','pmk')

676 BOOK PMK

665 NULL RMD

NULL DMO CHENNAI

123 DTP KOVAI

**3.foreign key constraint:**

NOTE: ONLy null value allowed

create table marks(mid int constraint fk\_mid foreign key references csccourse(cid),cname char(10))

insert into marks values(null,'tamil')

select \* from marks

NULL TAMIL

**4.check constraint :**

( condition value only allowed)

create table lib(lid int,lname char(10),age int constraint ck\_age check(age>=18))

insert into lib values(86675,'sasa',24)

34223 BALA 20

86675 SASA 24

**5.default constraint:**

create table emp3(ecode int,name char(15),city char(15),country char(15)constraint ck\_country default'india')

insert into emp3 values(86675,'sasa','chennai',default)

select \* from emp3

86675 SASA CHENNAI india

**chapter 5**

**i)views:**

View Is A Table It Consists Of Set Rows And Columns. It Comes From One Or More Tables (or) Other Views.

views types:

1. Updatable Views------>insert,select
2. Read Only Views------>only read

**creating views**

syntax:

create view viewname as select column name1, column name2... from tablename

ex:

create view eview as select ename, deptno from emp

**querying views:**

select \* from eview

**manipulating views:**

insert into viewname values(677,’ghhj’);

**alter views:**

alter view eview as select \* from emp where salary<=3000

drop views:

Drop view viewname

**2.indexes:**

two types of indexes:

1.clustered index

2.nonclustered index

**1.clustered index:(sort the column values)**

ex:Create clustered index in\_bname on book(name)

c++

sql

tally

**2.nonclustered index**:

Create nonclustered index in\_bkname on book(name)

**3.unique indexes:**

Create unique index in\_csccourse on csc2(course)

DCA

HDCA

DMO

ADJP

DTP

NULL

HDUA

C++

DUA

**4.composite indexes:**

Create index in\_csccourse on csc2(course,fees)

**5.droping indexes:**

Drop index book . in\_bkname

**sql server programming**

**chapter 1:**

**declare statement:**

used to create a local variables.

**set statement:**

to set or assign a value to a variable.

ex:

declare @x int, @y int ,@z int

set @x=100

set @y=50

set @z=@x\*@y

print @x

print @y

print 'Product of 2 numbers= '+convert(varchar,@z)

**output:**

100

50

Product of 2 numbers= 5000

**variables:**

**1.local variables**

they are declared in body of the batch or procedure using the declare statement

**2.GLOBAL VARIABLES**

i)select @@version

microsoft sql server 2000 - 8.00.194 (intel x86) aug 6 2000 00:57:48 copyright (c) 1988-2000 microsoft corporation developer edition on windows nt 5.1 (build 2600: service pack 3)

ii)select @@server name

SYSTEM-10

iii)select @@connections 24

**Temporary tables:**

**1.localtemporary variables:**

select \* into **#temp2**  from emp

select \* from #temp2

**2.global temporary variables:**

CREATE TABLE ##TEMPO2 (SNO INT,SNAME VARCHAR(10))

INSERT INTO ##TEMPO2 VALUES(5434,'BOOK')

SELECT \* FROM ##TEMPO2

**chapter 2:**

**2.1 statement blocks: begin........end**

a statement block is a series of block statements. if we need execute two or more sql statements within an IF or ELSE clause oe WHILE LOOP .

syntax:

BEGIN

{

(sql statement/statement block)

}

END

**2.2 CONDITIONAL PROCESSING:**

1. IF...ELSE Statement:

Syntax:

IF expression

{ statement/begin.....end}

ELSE

{statement/begin...end}]

Ex:1

select \* from emp

declare

@desig char(20)

select @desig =description from emp where empid=454

if @desig='doctor'

update emp set salary=salary+100 where empid=454

else if @desig='clerk'

update emp set salary=salary-300 where empid=454

else if @desig='teacher'

update emp set salary=salary+500 where empid=454

output:

454 clerk 45300.00 45.0000 NULL NULL

ex:2

declare

@x int

set @x=12

begin

if @x%2=0

print ‘an even number'

else

print ‘an odd number'

end

output:

an even number

**2.3 repetitive processing**:

1.while statement

declare

@x int,

set @x=0

while @x<=9

begin

select @x=@x+1

if @x%2=0

continue

print @x

end

**output:**

1

3

5

7

9