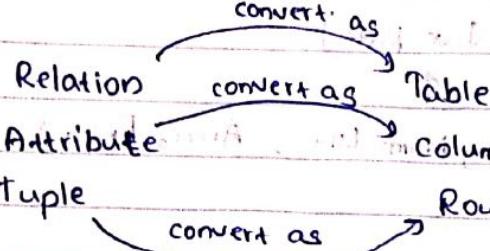


Lecture 04

SQL

SEQUEL C (Structured English Query Language)
standard version developed 1986

Relational model Vs SQL



Using SQL → Data Definition Language (DDL)
Sub set of SQL → Data Manipulation Language (DML)

Using SQL,

- ensure security & authorization
- Do transactions between tables and databases
- Facilitate connecting database with applications

Data Definition Language

Support to creation, deletion and modifications for tables and views.

Constraints

① Not Null → Ensure that column doesn't have null values.

② Unique → Ensure that all values in are different.
(C doesn't have duplicate values)

③ Primary key → Using Primary key uniquely identify values.

④ Foreign key → Handle data consistence between two tables

⑤ Default → Defines a default value for a column (When no values are given)

⑥ Check → Validates data in a column

Creating Table

Create Table Student

(

StudentId INTEGER PRIMARY KEY,
SName VARCHAR (30) NOT NULL,
nic CHAR (10) UNIQUE,
gpa FLOAT,
progid VARCHAR (10) DEFAULT 'IT',
CONSTRAINT student_prog_fk FOREIGN KEY (progid)
REFERENCES programs(id) ON DELETE SET DEFAULT
ON UPDATE CASCADE
CONSTRAINT gpa_ck CHECK (gpa <= 40)

)
on CONSTRAINT <> constraint name <> constraint type <> Reference
table / Condition / column <>

Modifications Tables.

- ① Adding new column to a table.

`ALTER TABLE student ADD age INT`

- ② Adding new constraint to a column

`ALTER TABLE student ADD CONSTRAINT chk_age CHECK (age > 18)`

- ③ Removing a column from a table

`ALTER TABLE student DROP COLUMN age`

- ④ Delete a table.

`DROP TABLE Employee`

Data Manipulation Language (DML)

Data

Manipulation involves

Inserting Data into the tables

Retrieving existing data

Deleting existing data from tables

Modifying data.

- ① Inserting a Row to a table

Insert values for all the columns ~~for~~ in the table.

`INSERT INTO <tablename> VALUES <values hope to insert>`
OR

`INSERT INTO <table name> (Column Name) VALUES <values hope to insert>`

Insert values for specific columns.

② Deleting arrow

Date: _____
If didn't update where clause
all the table will delete.

`DELETE <table name> WHERE < primarykey = value >`

③ Updating arrow

`UPDATE <table name>`

`SET < column name = value >`

`WHERE < primary key = exact column need to update >`

If didn't update where clause all the columns
will update ~~gpa~~

④ Retrieving information

SELECT CLAUSE

`SELECT < attributes >`

`FROM < one or more relations >` table that going to use

`WHERE < conditions >`

Ex. display ids of all students whose gpa is above 3.0

`SELECT StudentID`

`From Student`

`WHERE gpa > 3.0`

Like Operator

- Use to match the patterns

A% → Start with A

%A → End with A

A_ → Starting with A and after the A you have only one character. (2 letters)

IS [NOT] NULL Operator

→ Used to check whether attribute value is null

DISTINCT Operator

Remove duplicate values using distinct operator

Ex. Select progId
From Student

ProgId
BM
IT
SE
IT
BM

Select DISTINCT progId
From Student

ProgId
BM
IT
SE

Between Operator

→ Used to check whether attribute value is within a range.

Ex. Select StudentID

From Student

Where gpa between 3.7 and 4.00

ORDER BY CLAUSE

ASC

(Assending order)
(default)

DESC (Descending order)

Select Name, gpa
From Student
Order by gpa

Select Name, gpa

From Student
order by gpa DESC

[INNER] JOIN

Joins two tables base ~~on~~ on a certain condition

Steps

1. Need to find out common columns between tables
2. Match tables according to common columns.

Ex. Find the names of students who follow programs offered by SLIIT.

Select s.Name

From Student.s, program p
where s.pid = p.ProgId and
OfferedBy = 'SLIIT'

Student				Program		
SID	Name	GPA	(pid)	ProgID	years	Offered By
1000	Amal	3.2	BM	BM	3	Curtin
1001	Nimali	2.8	IT	IT	4	SLIIT
1002	Aruni	3.8	SE	SE	3	SHU
1003	Surani	2.5	IT	IT	3	

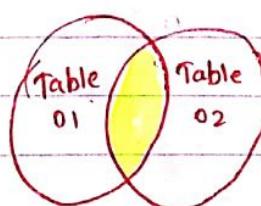
Name
Nimali
Surani

Select s.Name

From Student.s, INNER JOIN program.p on

s.pid = p.ProgId

Where OfferedBy = 'SLIIT'



Select < >
from <table1> inner join
<table2> on <table joining
condition>

Left Outer Join

Returns all records from the left table and with matching records in the right table.

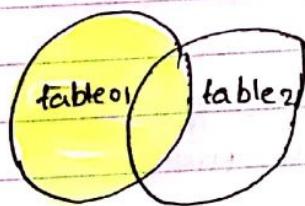
- The result is NULL in the right side when there is no match.

Ex. for all the students display the name and the offering institute

Select s.Name, p.offerBy
From Student s LEFT OUTER
JOIN program p
on s.pid = p.progId

Student				Program		
Std	Name	GPA	Pid	ProgId	Years	OfferBy
1000	Amal	3.2	BM	BM	3	Curtin
1001	Nimali	2.8	IT	IT	4	SLIIT
1002	Aruni	3.8	SE	null	null	null
1003	Surani	2.5	IT			

Left Join



Name	OfferBy
Amal	Curtin
Nimali	SLIIT
Aruni	NULL
Surani	SLIIT

Select [] Join
from <table1> left outer join <table2> on
<table joining condition> when right <table2> is not
qualified in left table

* result is zero in left side it is no match.

Right Outer Join

return all rows from the right hand side and matching rows in left table.

Student				Program			
SID	Name	GPA	Pid	ProgId	years	OfferBy	
1000	Amal	3.2	BM	BM	3	Curtin	
1001	Nimal	2.8	IT	IT	4	SLIIT	
1002	Aruni	3.0	NULL	SE	3	SHU	
1003	Surani	2.5	IT				

Ex. For all the programs display the offering institute and names of the students following.

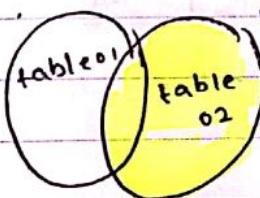
Name	OfferBy
Amal	Curtin
Nimal	SLIIT
Surani	SLIIT
NULL	SHU

Select S.Name , p.OfferBy

From Student S RIGHT OUTER JOIN program p

on s.pid = p.progid .

all the records in right with the matching rows of the left



Select
From <table1> right outer join <table2> on
<table joining condition> .

In Operator

Used to check whether attribute value matches any value within a value list

Ex. Find the students who has obtained a 'A'.

return
only name

```

Select s.Name
From Students
Where s.SID IN (Select SID
from Grades
Where Grade = 'A')

```

return
only the
grade

Student		
STD	Name	GPA
1000	Amal	3.2
1001	Nimali	2.8
1002	Aruni	3.8

Grades		
SID	cid	Grade
1000	IT102	A
1000	IT100	B
1001	IT102	A
1002	IT202	C
1002	IT200	C

Name
Amal
Nimali

Select

From

Where <column Name / Value> In (<list of value>)

Exists operator

Used to check if subquery returns any rows

- exist query can return true if the subquery return something.

Eg. Find the students who has obtained 'A'

Select S.
From Students

where EXISTS (Select*
from grades g
where g.SID = S.SID and
g.Grade = 'A')

Student		
SID	Name	GPA
1000	Amal	3.2
1001	Nimali	2.8
1002	Aruni	3.8

Grades		
SID	cid	Grade
1000	IT102	A
1000	IT100	B
1001	IT102	A
1002	IT102	C
1002	IT200	C

Name
Amal
Nimali

(Ans: Amal & Nimali)

Comparison Operators with Some, any & All

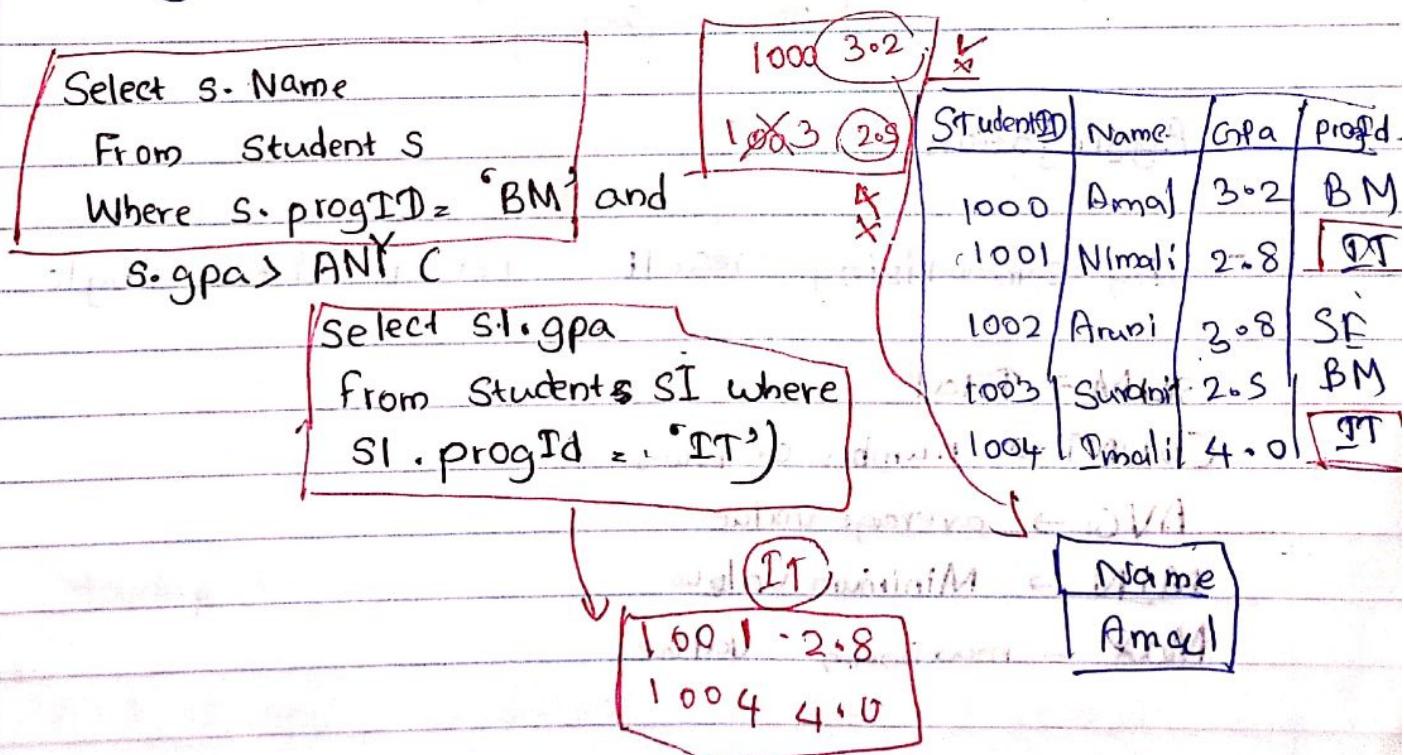
SOME and ANY,

- Used to compare a value to a list or Subquery.
- Return true if at least one of the comparison evaluates as true.

ALL:

- used to compare value to a list or Subquery.
- If all the expression evaluate to true then the result of the ALL expression will be true.
- Otherwise the result will be false.

Ex. Find BM Students who has gpa greater than any of the IT Students.



ALL Operator.

Ex. find IT students who has gpa greater than all the BM students.

Select s.Name
From Students
Where s.ProgID = 'IT' and
s.gpa > ALLC

Select SI.gpa
from Student SI where
SI.ProgID = 'BM'.

IT
1001 2.8 ✓
1004 4.0 ✓

Student

Student ID	Name	gpa	ProgID
1000	Amal	3.2	BM
1001	Nimali	2.8	IT
1002	Arunl	3.8	SE
1003	Suravnl	2.5	BM
1004	Ismali	4.0	IT

BM
1000 3.2
1003 2.5

Value list

Name
Email

Aggregation

- By summarizing result returning the a single value.

SUM → Total

COUNT → number of rows

AVG → average value

MIN → Minimum value

Max - maximum value.

Grouping (Grouping By Clause)

- group the data and provide single summary for each group
- Grouping is done based on values in a given field

Ex. Count the number of students who has followed each moduk.

Select CID, Count(SID)

From Student

Group by CID

Select sid, nic, convert(name)

From

where

group by sid, nic

order by

student	SID	CID	Grade
	1000	DBill	A
	1000	SET	B
	1001	DBill	A
	1002	DBill	C
	1002	SPD	C

	SID	CID	Grade
	1000	DBill	A
	1001	DBill	A
	1002	DBill	C
	1000	SET	B
	1002	SPD	C

CID	Count(SID)
DBill	3
SET	1
SPD	1

Having Clause

→ Used to apply conditions on the groupings.

Ex. Find courses which is followed by More than two students.

CID	Count(SID)
DBill	3

Select CID, Count(STID)

From course

Group by CID

Having count(STID) > 2

Step **SELECT <attribute-list>**

FROM <table-list>

[WHERE < condition>]

[GROUP BY <group attribute(s)>]

[HAVING <group condition>]

[ORDER BY & <attribute list>];