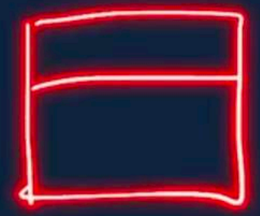

SQL
PHP
C++
HTML
CSS
C++
APNA
COLLEGE

Database

Database is collection of data in a format that can be easily accessed (Digital)

A software application used to manage our DB is called DBMS (Database Management System)

Types of Databases



Relational

Data stored in tables

MySQL · Microsoft SQL Server · ORACLE · PostgreSQL

Non-relational (NoSQL)

data not stored in tables

mongoDB

** We use SQL to work with relational DBMS

APNA COLLEGE

What is SQL?

Structured Query Language

SQL is a programming language used to interact with relational databases.

It is used to perform **CRUD** operations :

Create

Read

Update

Delete

SEQUEL

Structured

English

Query

Language

SQL

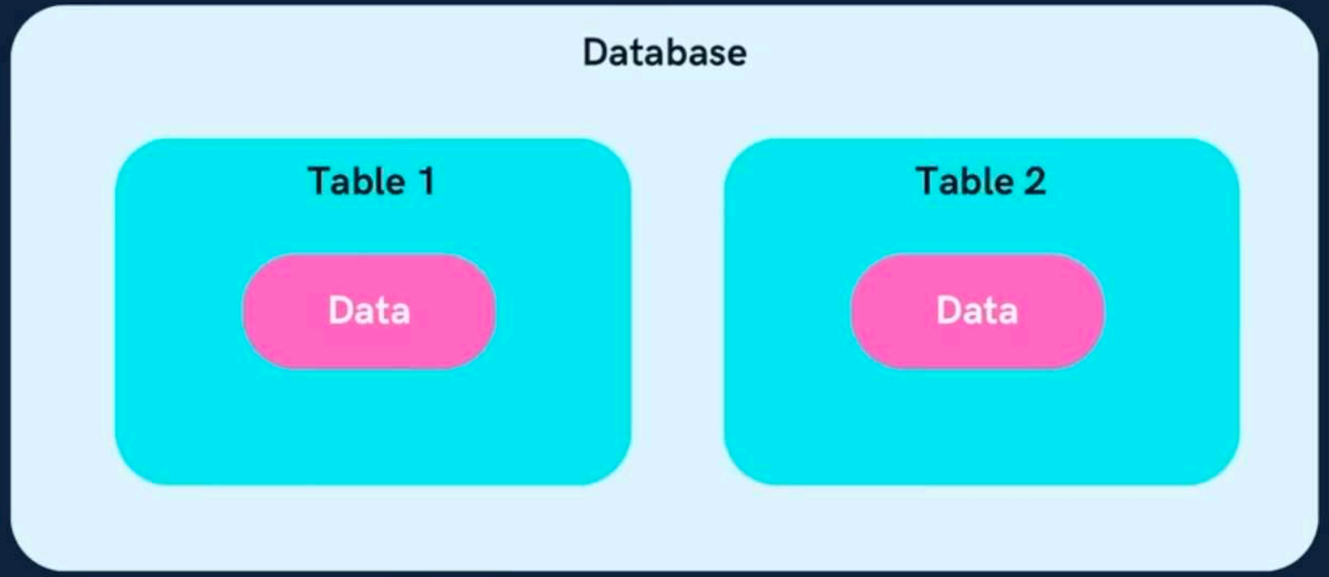
Structured

Query

Language

APNA COLLEGE

Database Structure



What is a table?

columns
rows

Student table

RollNo	Name	Class	DOB	Gender	City	Marks
1	Nanda	X	1995-06-06	M	Agra	551
2	Saurabh	XII	1993-05-07	M	Mumbai	462
3	Sonal	XI	1994-05-06	F	Delhi	400
4	Trisla	XII	1995-08-08	F	Mumbai	450
5	Store	XII	1995-10-08	M	Delhi	369
6	Marisla	XI	1994-12-12	F	Dubai	250
7	Neha	X	1995-12-08	F	Moscow	377
8	Nishant	X	1995-06-12	M	Moscow	489

row1
row2
row3

col1
col2
col3

APNA COLLEGE

Creating our First Database

Our first SQL Query

```
CREATE DATABASE db_name;
```

```
DROP DATABASE db_name;
```

Creating our First Table

```
USE db_name;

CREATE TABLE table_name (
    column_name1 datatype constraint,
    column_name2 datatype constraint,
    column_name2 datatype constraint
);
```

```
CREATE TABLE student (
  id INT PRIMARY KEY,
  name VARCHAR(50),
  age INT NOT NULL
);
```

APNA COLLEGE

Administration | Schemas | classroom*

SCHEMAS

Filter objects

- college
  - Tables
    - student
      - Columns
      - Indexes
      - Foreign Keys
      - Triggers
    - Views
    - Stored Procedures
    - Functions
- sys

Object Info | Session

No object selected

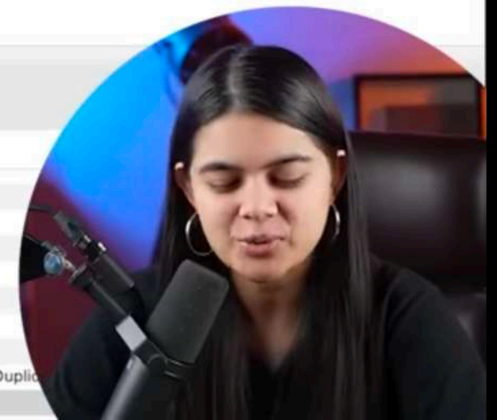
Limit to 1000 rows

```sql
CREATE DATABASE college;

USE college;

CREATE TABLE student (
  id INT PRIMARY KEY,
  name VARCHAR(50),
  age INT NOT NULL
);

INSERT INTO student VALUES(1, "AMAN", 26);
INSERT INTO student VALUES(2, "SHRADHA", 24);

SELECT * FROM student;
```

160% 23:14

Action Output

	Time	Action	Response
2	15:12:41	create database temp2	1 row(s) affected
3	15:13:26	CREATE DATABASE college	1 row(s) affected
4	15:14:06	DROP DATABASE temp1	0 row(s) affected
5	15:14:32	DROP DATABASE temp2	0 row(s) affected
6	15:15:29	USE college	0 row(s) affected
7	15:20:35	CREATE TABLE student (id INT PRIMARY KEY, name VARCHAR(50), age INT NOT NULL)	0 row(s) affected
8	15:22:39	INSERT INTO student VALUES(1, "AMAN", 26)	1 row(s) affected
9	15:22:44	INSERT INTO student VALUES(1, "SHRADHA", 24)	Error Code: 1062. Duplic
10	15:23:02	INSERT INTO student VALUES(2, "SHRADHA", 24)	1 row(s) affected

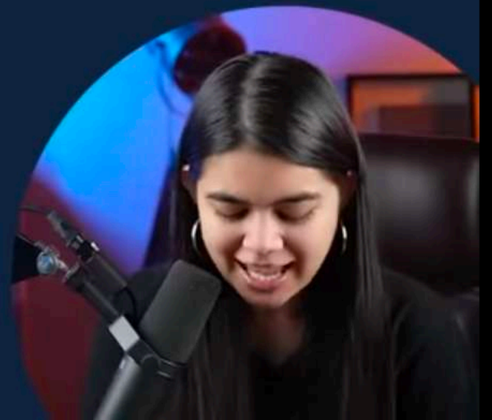


APNA COLLEGE

SQL Datatypes

They define the type of values that can be stored in a column

DATATYPE	DESCRIPTION	USAGE
CHAR	string(0-255), can store characters of fixed length	CHAR(50)
VARCHAR	string(0-255), can store characters up to given length	VARCHAR(50)
BLOB	string(0-65535), can store binary large object	BLOB(1000)
INT	integer(-2,147,483,648 to 2,147,483,647)	INT
TINYINT	integer(-128 to 127)	TINYINT
BIGINT	integer(-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807)	BIGINT
BIT	can store x-bit values. x can range from 1 to 64	BIT(2)
FLOAT	Decimal number - with precision to 23 digits	FLOAT
DOUBLE	Decimal number - with 24 to 53 digits	DOUBLE
BOOLEAN	Boolean values 0 or 1	BOOLEAN
DATE	date in format of YYYY-MM-DD ranging from 1000-01-01 to 9999-12-31	DATE
YEAR	year in 4 digits format ranging from 1901 to 2155	YEAR



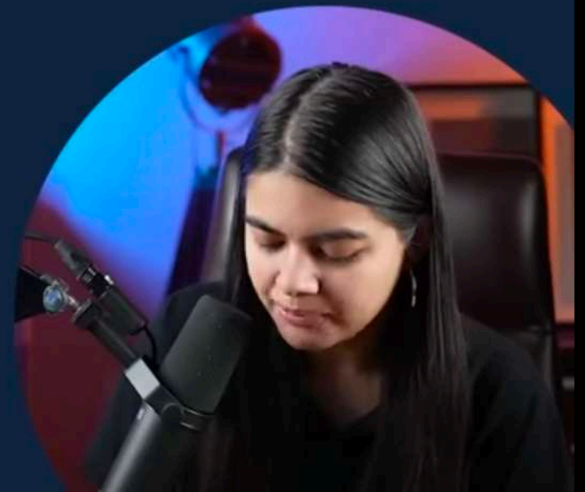
SQL Datatypes

Signed & Unsigned

numeric
↓
INT
FLOAT
DOUBLE
TIN`

TINYINT UNSIGNED (0 to 255)

TINYINT (-128 to 127)



APNA COLLEGE

Types of SQL Commands

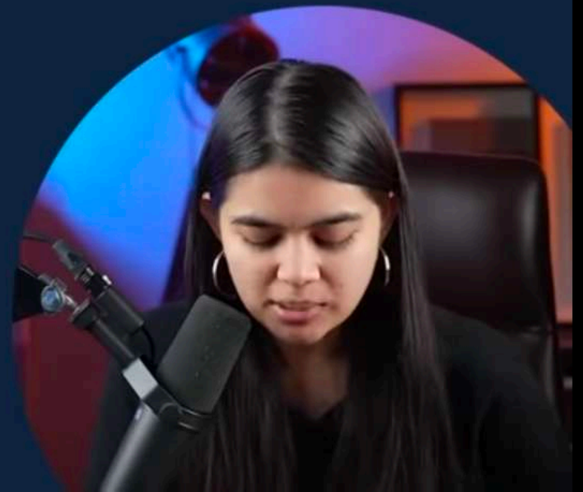
DDL (Data Definition Language) : create, alter, rename, truncate & drop

DQL (Data Query Language) : select

DML (Data Manipulation Language) : , insert, update & delete

DCL (Data Control Language) : grant & revoke permission to users

TCL (Transaction Control Language) : start transaction, commit, rollback e



Database related Queries

CREATE DATABASE *db_name;*

CREATE DATABASE IF NOT EXISTS *db_name;*

```
CREATE DATABASE IF NOT EXISTS college;
```

DROP DATABASE *db_name;*

DROP DATABASE IF EXISTS *db_name;*

SHOW DATABASES;

SHOW TABLES;

Table related Queries

Create

```
CREATE TABLE table_name (
   column_name1 datatype constraint,
   column_name2 datatype constraint,
);
```

```
CREATE TABLE student (
   rollno INT PRIMARY KEY,
   name VARCHAR(50)
);
```

Table related Queries

Select & View ALL columns

SELECT * FROM *table_name;*

```
SELECT * FROM student;
```

APNA COLLEGE

Table related Queries

Insert

INSERT INTO *table_name*
(colname1, colname2)
VALUES
(col1_v1, col2_v1),
(col1_v2, col2_v2);

```
INSERT INTO student
(rollno, name)
VALUES
(101, "karan"),
(102, "arjun");
```

Keys

Primary Key

It is a column (or set of columns) in a table that uniquely identifies each row. (a unique id)

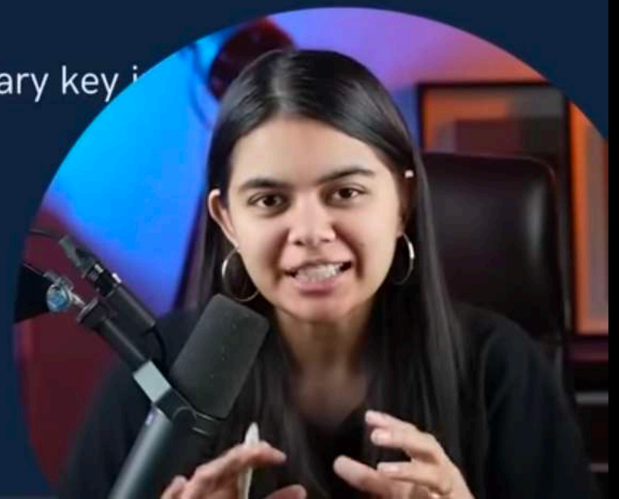
There is only 1 PK & it should be NOT null.

Foreign Key

A foreign key is a column (or set of columns) in a table that refers to the primary key i

There can be multiple FKs.

FKs can have duplicate & null values.



Keys

table1 - Student

id	name	cityId	city
101	karan	1	Pune
102	arjun	2	Mumbai
103	ram	1	Pune
104	shyam	3	Delhi

table2 - City

id	city_name
1	Pune
2	Mumbai
3	Delhi

APNA COLLEGE

Constraints

SQL constraints are used to specify rules for data in a table.

NOT NULL columns cannot have a null value

```
col1 int NOT NULL
```

UNIQUE all values in column are different

```
col2 int UNIQUE
```

PRIMARY KEY makes a column unique & not null but used only for one

```
id int PRIMARY KEY
```

```
CREATE TABLE temp (
  id int not null,
  PRIMARY KEY (id)
);
```

APNA COLLEGE

Administration | Schemas | classroom*

SCHEMAS

Filter objects

- college
  - Tables
    - student
    - temp1
  - Views
  - Stored Procedures
  - Functions
- sys

Object Info | Session

Table: temp1

Columns:
id int

Limit to 1000 rows

```
10
11 • CREATE TABLE temp1 (
12     id INT UNIQUE
13 );
14
15 • INSERT INTO temp1 VALUES (101);
16 • INSERT INTO temp1 VALUES (101);
17
18 • SELECT * FROM temp1;
19
20
21
22
23
24
25
```

160% | 24:16

Action Output

	Time	Action	Respo
27	15:56:59	CREATE TABLE employee(id INT PRIMARY KEY, name VARCHAR(100), salary INT)	0 row
28	15:58:15	INSERT INTO employee (id, name, salary) VALUES (1, "adam", 25000), (2, "bob", 30000), (3, "casey", 40000)	3 row
29	15:58:36	SELECT * FROM employee LIMIT 0, 1000	3 row
30	16:17:51	DROP database xyz_company	1 row
31	16:17:58	USE college	0 row
32	16:26:27	CREATE TABLE temp1 (id INT UNIQUE)	0 row
33	16:27:00	INSERT INTO temp1 VALUES (101)	1 row(s)
34	16:27:12	SELECT * FROM temp1 LIMIT 0, 1000	1 row(s)
35	16:27:19	INSERT INTO temp1 VALUES (101)	Error Cod

APNA COLLEGE

Administration | Schemas | classroom*

SCHEMAS

Filter objects

- college
  - Tables
    - student
    - temp1
  - Views
  - Stored Procedures
  - Functions
- sys

Object Info | Session

Table: temp1

Columns:
id int

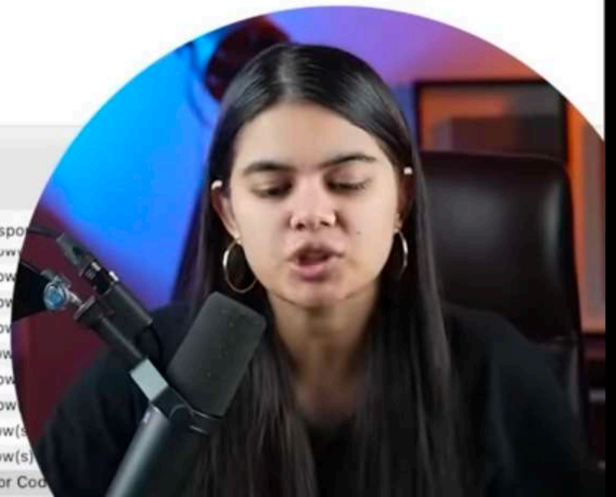
Limit to 1000 rows

```
9  SELECT * FROM student;
10
11 CREATE TABLE temp1 (
12  id INT,
13  name VARCHAR(50),
14  age INT,
15  city VARCHAR(20),
16  PRIMARY KEY (id, name)
17 );
18
19
20
21
22
23
24
```

160% 24:16

Action Output

	Time	Action	Response
27	15:56:59	CREATE TABLE employee(id INT PRIMARY KEY, name VARCHAR(100), salary INT)	0 row
28	15:58:15	INSERT INTO employee (id, name, salary) VALUES (1, "adam", 25000), (2, "bob", 30000), (3, "casey", 40000)	3 row
29	15:58:36	SELECT * FROM employee LIMIT 0, 1000	3 row
30	16:17:51	DROP database xyz_company	1 row
31	16:17:58	USE college	0 row
32	16:26:27	CREATE TABLE temp1 (id INT UNIQUE)	0 row
33	16:27:00	INSERT INTO temp1 VALUES (101)	1 row(s)
34	16:27:12	SELECT * FROM temp1 LIMIT 0, 1000	1 row(s)
35	16:27:19	INSERT INTO temp1 VALUES (101)	Error Cod



APNA COLLEGE

Constraints

FOREIGN KEY prevent actions that would destroy links between tables

```
CREATE TABLE temp (
  cust_id int,
  FOREIGN KEY (cust_id) references customer(id)
);
```

DEFAULT sets the default value of a column

```
salary INT DEFAULT 25000
```

Administration | Schemas | classroom*

SCHEMAS

Filter objects

- college
  - Tables
    - student
    - temp1
  - Views
  - Stored Procedures
  - Functions
- sys

Object Info | Session

Table: temp1

Columns:
id int

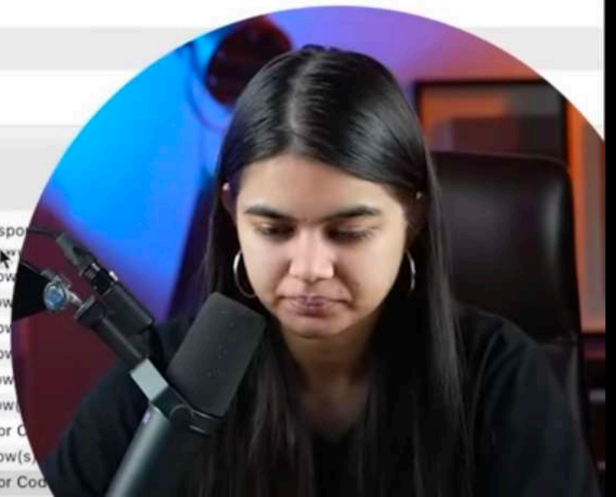
Limit to 1000 rows

```
10
11 • CREATE TABLE temp1 (
12     id INT,
13     name VARCHAR(50),
14     age INT,
15     city VARCHAR(20),
16     PRIMARY KEY (id, name)
17 );
18
19
20 • CREATE TABLE emp (
21       id INT,
22       salary INT DEFAULT 25000);
23
24   INSERT INTO (id) emp VALUES (101);
25
```

160% | 1:24 | 1 error found

Action Output

	Time	Action	Response
29	15:58:36	SELECT * FROM employee LIMIT 0, 1000	3 row
30	16:17:51	DROP database xyz_company	1 row
31	16:17:58	USE college	0 row
32	16:26:27	CREATE TABLE temp1 (id INT UNIQUE)	0 row
33	16:27:00	INSERT INTO temp1 VALUES (101)	1 row
34	16:27:12	SELECT * FROM temp1 LIMIT 0, 1000	1 row
35	16:27:19	INSERT INTO temp1 VALUES (101)	Error C
36	16:33:45	CREATE TABLE emp (id INT, salary INT DEFAULT 25000)	0 row(s)
37	16:34:07	INSERT INTO (id) emp VALUES (101)	Error Cod



APNA COLLEGE

Constraints

CHECK it can limit the values allowed in a column

```sql
CREATE TABLE city (
  id INT PRIMARY KEY,
  city VARCHAR(50),
  age INT,
  CONSTRAINT age_check CHECK (age >= 18 AND city="Delhi")
);
```

```sql
CREATE TABLE newTab (
  age INT CHECK (age >= 18)
);
```

APNA COLLEGE

Create this sample table

```sql
CREATE DATABASE college;
USE college;

CREATE TABLE student (
   rollno INT PRIMARY KEY,
   name VARCHAR(50),
   marks INT NOT NULL,
   grade VARCHAR(1),
   city VARCHAR(20)
);
```

Insert this data

```sql
INSERT INTO student
(rollno, name, marks, grade, city)
VALUES
(101, "anil", 78, "C", "Pune"),
(102, "bhumika", 93, "A", "Mumbai"),
(103, "chetan", 85, "B", "Mumbai"),
(104, "dhruv", 96, "A", "Delhi"),
(105, "emanuel", 12, "F", "Delhi"),
(106, "farah", 82, "B", "Delhi");
```

APNA COLLEGE

Local instance 3306 - Warning - not supported

Administration | Schemas

SCHEMAS

Filter objects

- college
- sys

Object Info | Session

No object selected

classroom*

Limit to 1000 rows

```sql
CREATE TABLE student (
    rollno INT PRIMARY KEY,
    name VARCHAR(50),
    marks INT NOT NULL,
    grade VARCHAR(1),
    city VARCHAR(20)
);

INSERT INTO student
(rollno, name, marks, grade, city)
VALUES
(101, "anil", 78, "C", "Pune"),
(102, "bhumika", 93, "A", "Mumbai"),
(103, "chetan", 85, "B", "Mumbai"),
(104, "dhruv", 96, "A", "Delhi"),
(105, "emanuel", 12, "F", "Delhi"),
(106, "farah", 82, "B", "Delhi");
```

160% 34:20

Action Output

	Time	Action	Resp
1	13:23:34	CREATE DATABASE college	1 row(
2	13:23:42	USE college	0 row(s
3	13:23:48	CREATE TABLE student (rollno INT PRIMARY KEY, name VARCHAR(50), marks INT NOT NULL, grade VARCHAR(1)...	0 row(s)
4	13:23:55	INSERT INTO student (rollno, name, marks, grade, city) VALUES (101, "anil", 78, "C", "Pune"), (102, "bhumika", 93, "A",...	6 row(s) a

APNA COLLEGE

Select in Detail

used to select any data from the database

Basic Syntax

```
SELECT col1, col2 FROM table_name;
```

To Select ALL

```
SELECT * FROM table_name;
```

Administration | Schemas | classroom*

SCHEMAS

Filter objects

- college
  - Tables
  - Views
  - Stored Procedures
  - Functions
- sys

Object Info | Session

No object selected

Limit to 1000 rows

```
  marks INT NOT NULL,
  grade VARCHAR(1),
  city VARCHAR(20)
);

INSERT INTO student
(rollno, name, marks, grade, city)
VALUES
(101, "anil", 78, "C", "Pune"),
(102, "bhumika", 93, "A", "Mumbai"),
(103, "chetan", 85, "B", "Mumbai"),
(104, "dhruv", 96, "A", "Delhi"),
(105, "emanuel", 12, "F", "Delhi"),
(106, "farah", 82, "B", "Delhi");

SELECT name, marks FROM student;
```

160% 20:22

Action Output

	Time	Action	Resp
1	13:23:34	CREATE DATABASE college	1 row
2	13:23:42	USE college	0 row
3	13:23:48	CREATE TABLE student (rollno INT PRIMARY KEY, name VARCHAR(50), marks INT NOT NULL, grade VARCHAR(1)...	0 row(s)
4	13:23:55	INSERT INTO student (rollno, name, marks, grade, city) VALUES (101, "anil", 78, "C", "Pune"), (102, "bhumika", 93, "A",...	6 row(s)
5	13:25:04	SELECT name, marks FROM student LIMIT 0, 1000	6 row(s)

APNA COLLEGE

Administration | Schemas | classroom*

SCHEMAS

Filter objects

- college
  - Tables
  - Views
  - Stored Procedures
  - Functions
- sys

Object Info | Session

No object selected

Limit to 1000 rows

```
  marks INT NOT NULL,
  grade VARCHAR(1),
  city VARCHAR(20)
);

INSERT INTO student
(rollno, name, marks, grade, city)
VALUES
(101, "anil", 78, "C", "Pune"),
(102, "bhumika", 93, "A", "Mumbai"),
(103, "chetan", 85, "B", "Mumbai"),
(104, "dhruv", 96, "A", "Delhi"),
(105, "emanuel", 12, "F", "Delhi"),
(106, "farah", 82, "B", "Delhi");

SELECT DISTINCT city FROM student;
```

160% 36:22

Action Output

	Time	Action	Resp
1	13:23:34	CREATE DATABASE college	1 row
2	13:23:42	USE college	0 row
3	13:23:48	CREATE TABLE student (rollno INT PRIMARY KEY, name VARCHAR(50), marks INT NOT NULL, grade VARCHAR(1)...	0 row(s)
4	13:23:55	INSERT INTO student (rollno, name, marks, grade, city) VALUES (101, "anil", 78, "C", "Pune"), (102, "bhumika", 93, "A",...	6 row(s)
5	13:25:04	SELECT name, marks FROM student LIMIT 0, 1000	6 row(s)

APNA COLLEGE

Where Clause

To define some conditions

SELECT *col1, col2* FROM *table_name*
WHERE *conditions;*

```
SELECT * FROM student WHERE marks > 80;
SELECT * FROM student WHERE city = "Mumbai";
```

Administration | Schemas | classroom*

SCHEMAS

Filter objects

- college
  - Tables
  - Views
  - Stored Procedures
  - Functions
- sys

Object Info | Session

No object selected

Limit to 1000 rows

```
11
12 • INSERT INTO student
13   (rollno, name, marks, grade, city)
14   VALUES
15   (101, "anil", 78, "C", "Pune"),
16   (102, "bhumika", 93, "A", "Mumbai"),
17   (103, "chetan", 85, "B", "Mumbai"),
18   (104, "dhruv", 96, "A", "Delhi"),
19   (105, "emanuel", 12, "F", "Delhi"),
20   (106, "farah", 82, "B", "Delhi");
21
22 • SELECT DISTINCT city FROM student;
23
24 • SELECT *
25   FROM student
26   WHERE marks > 80 AND city = "Mumbai";
27
```

160% 21:26

Action Output

	Time	Action	Resp
5	13:25:04	SELECT name, marks FROM student LIMIT 0, 1000	6 row
6	13:25:47	SELECT * FROM student LIMIT 0, 1000	6 row
7	13:26:29	SELECT city FROM student LIMIT 0, 1000	6 row(s)
8	13:26:51	SELECT DISTINCT city FROM student LIMIT 0, 1000	3 row(s)
9	13:28:51	SELECT * FROM student WHERE marks > 80 LIMIT 0, 1000	4 row(s)

APNA COLLEGE

Where Clause

Using Operators in WHERE

Arithmetic Operators : +(addition) , -(subtraction), *(multiplication), /(division), %(modulus)

Comparison Operators : = (equal to), != (not equal to), > , >=, <, <=

Logical Operators : AND, OR , NOT, IN, BETWEEN, ALL, LIKE, ANY

Bitwise Operators : & (Bitwise AND), | (Bitwise OR)

Administration | Schemas | classroom*

SCHEMAS

Filter objects

- college
  - Tables
  - Views
  - Stored Procedures
  - Functions
- sys

Object Info | Session

No object selected

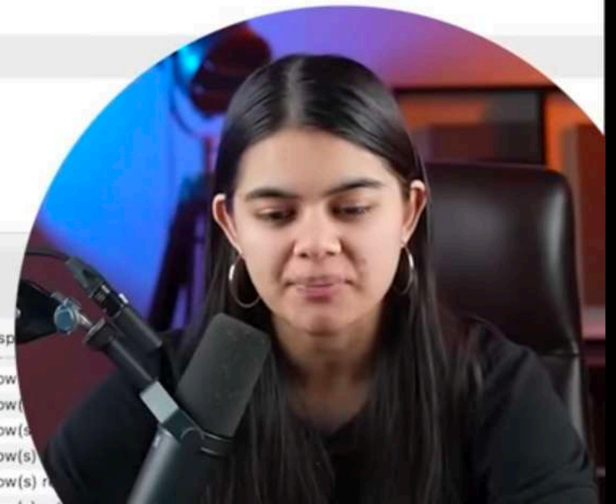
Limit to 1000 rows

```
11
12 • INSERT INTO student
13   (rollno, name, marks, grade, city)
14   VALUES
15   (101, "anil", 78, "C", "Pune"),
16   (102, "bhumika", 93, "A", "Mumbai"),
17   (103, "chetan", 85, "B", "Mumbai"),
18   (104, "dhruv", 96, "A", "Delhi"),
19   (105, "emanuel", 12, "F", "Delhi"),
20   (106, "farah", 82, "B", "Delhi");
21
22 • SELECT DISTINCT city FROM student;
23
24 • SELECT *
25   FROM student
26   WHERE marks+10 > 100 ;
27
```

160% 1:24

Action Output

	Time	Action	Resp
5	13:25:04	SELECT name, marks FROM student LIMIT 0, 1000	6 row
6	13:25:47	SELECT * FROM student LIMIT 0, 1000	6 row
7	13:26:29	SELECT city FROM student LIMIT 0, 1000	6 row
8	13:26:51	SELECT DISTINCT city FROM student LIMIT 0, 1000	3 row(s)
9	13:28:51	SELECT * FROM student WHERE marks > 80 LIMIT 0, 1000	4 row(s)



APNA COLLEGE

Operators

AND (to check for both conditions to be true)

```
SELECT * FROM student WHERE marks > 80 AND city = "Mumbai";
```

OR (to check for one of the conditions to be true)

```
SELECT * FROM student WHERE marks > 90 OR city = "Mumbai";
```

APNA COLLEGE

Operators

Between (selects for a given range)

```
SELECT * FROM student WHERE marks BETWEEN 80 AND 90;
```

In (matches any value in the list)

```
SELECT * FROM student WHERE city IN ("Delhi", "Mumbai");
```

NOT (to negate the given condition)

```
SELECT * FROM student WHERE city NOT IN ("Delhi", "Mumba
```

Limit Clause

Sets an upper limit on number of (tuples)rows to be returned

```
SELECT * FROM student LIMIT 3;
```

```
SELECT col1, col2 FROM table_name
LIMIT number;
```

Order By Clause

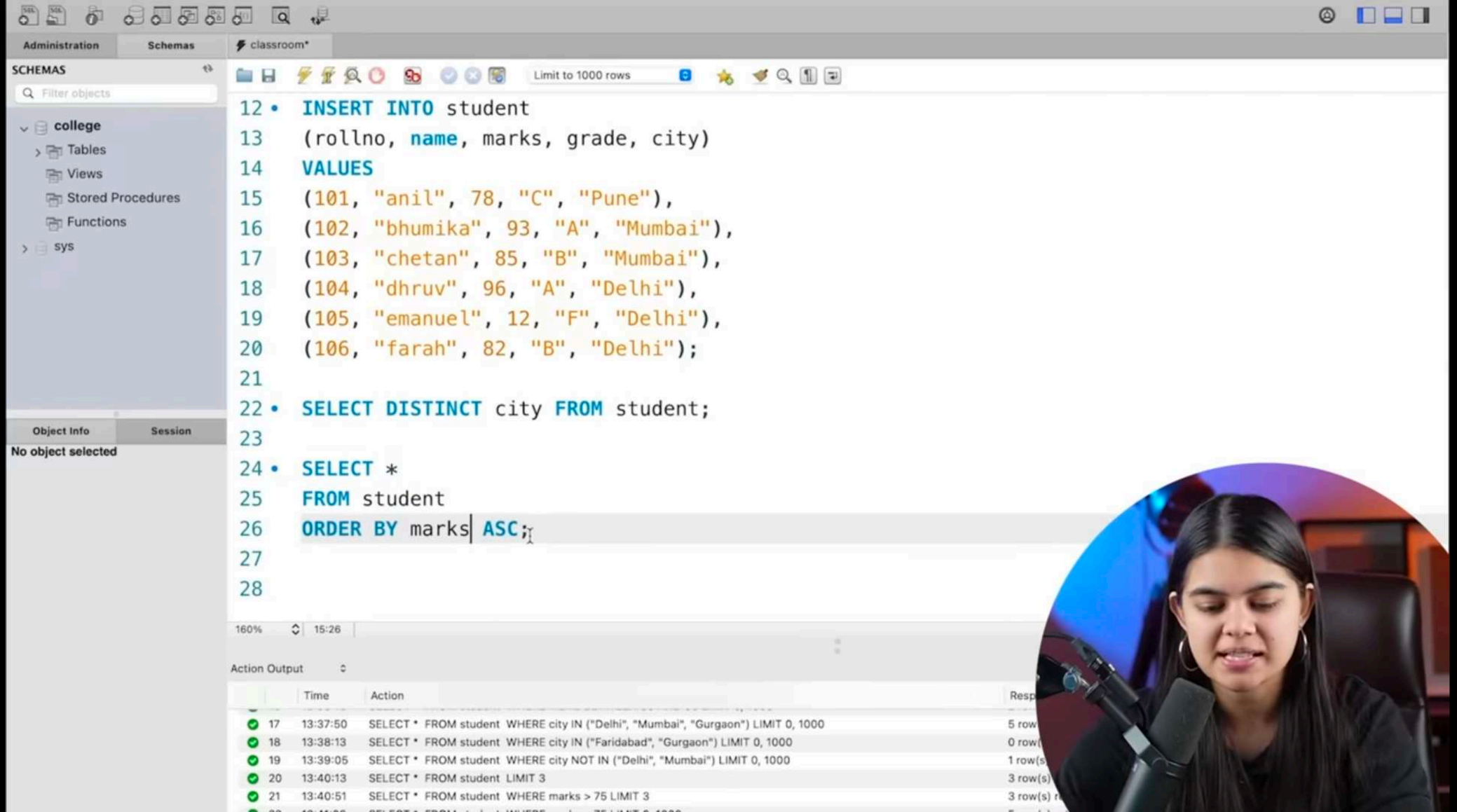
1 2 3 4 5 →

To sort in ascending (ASC) or descending order (DESC)

```
SELECT * FROM student
ORDER BY city ASC;
```

```
SELECT col1, col2 FROM table_name
ORDER BY col_name(s) ASC;
```

APNA COLLEGE


Administration
Schemas
classroom*
SCHEMAS
Filter objects
college
Tables
Views
Stored Procedures
Functions
sys
Object Info
Session
No object selected
Limit to 1000 rows
12 • INSERT INTO student
13 (rollno, name, marks, grade, city)
14 VALUES
15 (101, "anil", 78, "C", "Pune"),
16 (102, "bhumika", 93, "A", "Mumbai"),
17 (103, "chetan", 85, "B", "Mumbai"),
18 (104, "dhruv", 96, "A", "Delhi"),
19 (105, "emanuel", 12, "F", "Delhi"),
20 (106, "farah", 82, "B", "Delhi");
21
22 • SELECT DISTINCT city FROM student;
23
24 • SELECT *
25 FROM student
26 ORDER BY marks ASC;
27
28
160%
15:26
Action Output
Time
Action
17 13:37:50 SELECT * FROM student WHERE city IN ("Delhi", "Mumbai", "Gurgaon") LIMIT 0, 1000
18 13:38:13 SELECT * FROM student WHERE city IN ("Faridabad", "Gurgaon") LIMIT 0, 1000
19 13:39:05 SELECT * FROM student WHERE city NOT IN ("Delhi", "Mumbai") LIMIT 0, 1000
20 13:40:13 SELECT * FROM student LIMIT 3
21 13:40:51 SELECT * FROM student WHERE marks > 75 LIMIT 3

Administration | Schemas | classroom*

SCHEMAS

Filter objects

- college
  - Tables
  - Views
  - Stored Procedures
  - Functions
- sys

Object Info | Session

No object selected

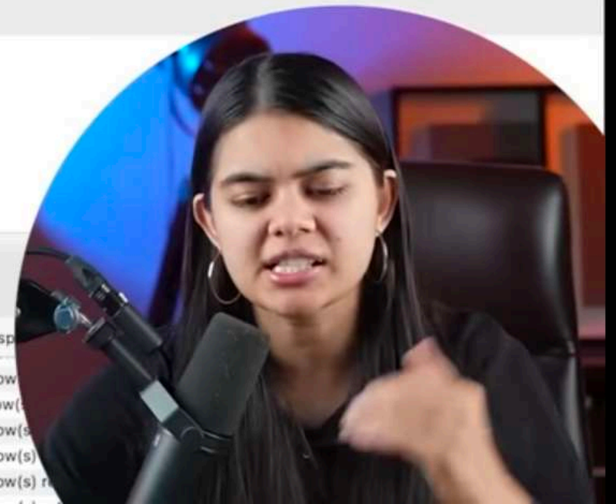
Limit to 1000 rows

```
12 • INSERT INTO student
13   (rollno, name, marks, grade, city)
14   VALUES
15   (101, "anil", 78, "C", "Pune"),
16   (102, "bhumika", 93, "A", "Mumbai"),
17   (103, "chetan", 85, "B", "Mumbai"),
18   (104, "dhruv", 96, "A", "Delhi"),
19   (105, "emanuel", 12, "F", "Delhi"),
20   (106, "farah", 82, "B", "Delhi");
21
22 • SELECT DISTINCT city FROM student;
23
24 • SELECT *
25   FROM student
26   ORDER BY marks DESC;
27
28
```

160% 1:24

Action Output

	Time	Action	Resp
18	13:38:13	SELECT * FROM student WHERE city IN ("Faridabad", "Gurgaon") LIMIT 0, 1000	0 row
19	13:39:05	SELECT * FROM student WHERE city NOT IN ("Delhi", "Mumbai") LIMIT 0, 1000	1 row(
20	13:40:13	SELECT * FROM student LIMIT 3	3 row(
21	13:40:51	SELECT * FROM student WHERE marks > 75 LIMIT 3	3 row(s)
22	13:41:03	SELECT * FROM student WHERE marks > 75 LIMIT 0, 1000	5 row(s) r



APNA COLLEGE

Aggregate Functions

Aggregare functions perform a calculation on a set of values, and return a single value.

- COUNT()
- MAX()
- MIN()
- SUM()
- AVG()

Get Maximum Marks

```
SELECT max(marks)
FROM student;
```

Get Average marks

```
SELECT avg(marks)
FROM student;
```

Administration | Schemas | classroom*

SCHEMAS

Filter objects

- college
  - Tables
  - Views
  - Stored Procedures
  - Functions
- sys

Object Info | Session

No object selected

Limit to 1000 rows

```
11
12 • INSERT INTO student
13   (rollno, name, marks, grade, city)
14   VALUES
15   (101, "anil", 78, "C", "Pune"),
16   (102, "bhumika", 93, "A", "Mumbai"),
17   (103, "chetan", 85, "B", "Mumbai"),
18   (104, "dhruv", 96, "A", "Delhi"),
19   (105, "emanuel", 12, "F", "Delhi"),
20   (106, "farah", 82, "B", "Delhi");
21
22 • SELECT DISTINCT city FROM student;
23
24 • SELECT MAX(marks)
25   FROM student;
26
27
```

160% 1:24

Action Output

	Time	Action	Resp
21	13:40:51	SELECT * FROM student WHERE marks > 75 LIMIT 3	3 row
22	13:41:03	SELECT * FROM student WHERE marks > 75 LIMIT 0, 1000	5 row
23	13:42:56	SELECT * FROM student ORDER BY city ASC LIMIT 0, 1000	6 row(s)
24	13:43:10	SELECT * FROM student ORDER BY marks ASC LIMIT 0, 1000	6 row(s)
25	13:43:42	SELECT * FROM student ORDER BY marks DESC LIMIT 0, 1000	6 row(s)

APNA COLLEGE

Group By Clause

Groups rows that have the same values into summary rows.
It collects data from multiple records and groups the result by one or more column.

*Generally we use group by with some *aggregation function*.

Count number of students in each city

```
SELECT city, count(name)
FROM student
GROUP BY city;
```

Administration | Schemas | classroom*

SCHEMAS

Filter objects

- college
  - Tables
  - Views
  - Stored Procedures
  - Functions
- sys

Object Info | Session

No object selected

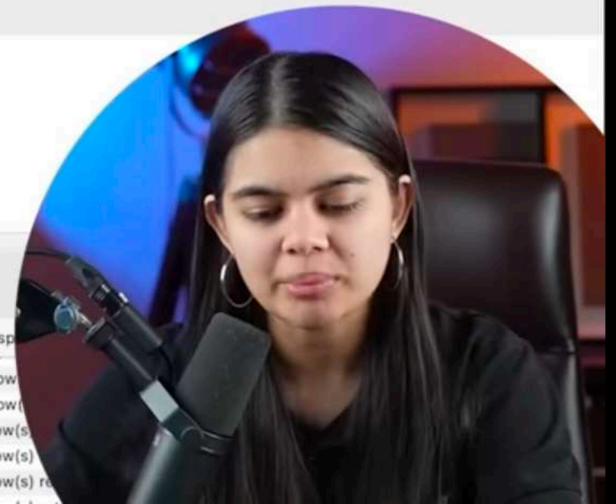
Limit to 1000 rows

```
12 • INSERT INTO student
13   (rollno, name, marks, grade, city)
14   VALUES
15   (101, "anil", 78, "C", "Pune"),
16   (102, "bhumika", 93, "A", "Mumbai"),
17   (103, "chetan", 85, "B", "Mumbai"),
18   (104, "dhruv", 96, "A", "Delhi"),
19   (105, "emanuel", 12, "F", "Delhi"),
20   (106, "farah", 82, "B", "Delhi");
21
22 • SELECT DISTINCT city FROM student;
23
24 • SELECT city, count(rollno)
25   FROM student
26   GROUP BY city;
27
28
```

160% 12:24

Action Output

	Time	Action	Resp
26	13:44:07	SELECT * FROM student ORDER BY marks DESC LIMIT 3	3 row
27	13:47:14	SELECT marks FROM student LIMIT 0, 1000	6 row
28	13:47:50	SELECT MAX(marks) FROM student LIMIT 0, 1000	1 row(s)
29	13:48:03	SELECT MIN(marks) FROM student LIMIT 0, 1000	1 row(s)
30	13:48:14	SELECT AVG(marks) FROM student LIMIT 0, 1000	1 row(s)



APNA COLLEGE

Administration | Schemas

SCHEMAS

Filter objects

- college
  - Tables
  - Views
  - Stored Procedures
  - Functions
- sys

Object Info | Session

No object selected

classroom*

Limit to 1000 rows

```
12 • INSERT INTO student
13   (rollno, name, marks, grade, city)
14   VALUES
15   (101, "anil", 78, "C", "Pune"),
16   (102, "bhumika", 93, "A", "Mumbai"),
17   (103, "chetan", 85, "B", "Mumbai"),
18   (104, "dhruv", 96, "A", "Delhi"),
19   (105, "emanuel", 12, "F", "Delhi"),
20   (106, "farah", 82, "B", "Delhi");
21
22 • SELECT DISTINCT city FROM student;
23
24 • SELECT city,name, count(rollno)
25   FROM student
26   GROUP BY city, name;
27
28
```

160% 1:24

Action Output

	Time	Action	Resp
28	13:47:50	SELECT MAX(marks) FROM student LIMIT 0, 1000	1 row
29	13:48:03	SELECT MIN(marks) FROM student LIMIT 0, 1000	1 row
30	13:48:14	SELECT AVG(marks) FROM student LIMIT 0, 1000	1 row
31	13:48:37	SELECT COUNT(rollno) FROM student LIMIT 0, 1000	1 row
32	13:51:46	SELECT city FROM student GROUP BY city LIMIT 0, 1000	3 row
33	13:52:17	SELECT city, count(rollno) FROM student GROUP BY city LIMIT 0, 1000	3 row(s)
34	13:53:32	SELECT city,name, count(rollno) FROM student GROUP BY city LIMIT 0, 1000	Error Cod

APNA COLLEGE

Having Clause

Similar to Where i.e. applies some condition on rows.
Used when we want to apply any condition after grouping.

Count number of students in each city where max marks cross 90.

```
SELECT count(name), city
FROM student
GROUP BY city
HAVING max(marks) > 90;
```

General Order

```
SELECT column(s)
FROM table_name
WHERE condition
GROUP BY column(s)
HAVING condition
ORDER BY column(s) ASC;
```

General Order

```
SELECT column(s)
FROM table_name
WHERE condition
GROUP BY column(s)
HAVING condition
ORDER BY column(s) ASC;
```

where → rows

having → groups

Administration | Schemas

SCHEMAS

Filter objects

- college
  - Tables
  - Views
  - Stored Procedures
  - Functions
- sys

Object Info | Session

No object selected

classroom*

Limit to 1000 rows

```
13  (rollno, name, marks, grade, city)
14  VALUES
15  (101, "anil", 78, "C", "Pune"),
16  (102, "bhumika", 93, "A", "Mumbai"),
17  (103, "chetan", 85, "B", "Mumbai"),
18  (104, "dhruv", 96, "A", "Delhi"),
19  (105, "emanuel", 12, "F", "Delhi"),
20  (106, "farah", 82, "B", "Delhi");
21
22  SELECT DISTINCT city FROM student;
23
24  SELECT city
25  FROM student
26  WHERE grade = "A"
27  GROUP BY city
28  HAVING MAX(marks) > 93;
29
```

160% 30:16

Action Output

	Time	Action	Resp
42	14:03:21	SELECT grade FROM student GROUP BY grade ORDER BY grade LIMIT 0, 1000	4 row
43	14:03:38	SELECT grade, count(rollno) FROM student GROUP BY grade ORDER BY grade LIMIT 0, 1000	4 row
44	14:06:15	SELECT city FROM student GROUP BY city LIMIT 0, 1000	3 row
45	14:06:28	SELECT city, count(rollno) FROM student GROUP BY city LIMIT 0, 1000	3 row
46	14:07:01	SELECT city, count(rollno) FROM student GROUP BY city HAVING MAX(marks) > 90 LIMIT 0, 1000	2 row
47	14:11:04	SELECT city FROM student WHERE grade = "A" GROUP BY city LIMIT 0, 1000	2 row(s)
48	14:11:46	SELECT city FROM student WHERE grade = "A" GROUP BY city HAVING MAX(marks) > 93 LIMIT 0, 1000	1 row(s)

APNA COLLEGE

Administration | Schemas

SCHEMAS

Filter objects

- college
  - Tables
  - Views
  - Stored Procedures
  - Functions
- sys

Object Info | Session

No object selected

classroom*

Limit to 1000 rows

```
14  VALUES
15  (101, "anil", 78, "C", "Pune"),
16  (102, "bhumika", 93, "A", "Mumbai"),
17  (103, "chetan", 85, "B", "Mumbai"),
18  (104, "dhruv", 96, "A", "Delhi"),
19  (105, "emanuel", 12, "F", "Delhi"),
20  (106, "farah", 82, "B", "Delhi");
21
22  SELECT DISTINCT city FROM student;
23
24  SELECT city
25  FROM student
26  WHERE grade = "A"
27  GROUP BY city
28  HAVING MAX(marks) >= 93
29  ORDER BY ASC;
30
```

160% | 14:29 | 1 error found

Action Output

	Time	Action	Resp
43	14:03:38	SELECT grade, count(rollno) FROM student GROUP BY grade ORDER BY grade LIMIT 0, 1000	4 row
44	14:06:15	SELECT city FROM student GROUP BY city LIMIT 0, 1000	3 row
45	14:06:28	SELECT city, count(rollno) FROM student GROUP BY city LIMIT 0, 1000	3 row
46	14:07:01	SELECT city, count(rollno) FROM student GROUP BY city HAVING MAX(marks) > 90 LIMIT 0, 1000	2 row
47	14:11:04	SELECT city FROM student WHERE grade = "A" GROUP BY city LIMIT 0, 1000	2 row
48	14:11:46	SELECT city FROM student WHERE grade = "A" GROUP BY city HAVING MAX(marks) > 93 LIMIT 0, 1000	1 row(s)
49	14:11:58	SELECT city FROM student WHERE grade = "A" GROUP BY city HAVING MAX(marks) >= 93 LIMIT 0, 1000	2 row(s)

APNA COLLEGE

Administration | Schemas

SCHEMAS

Filter objects

- college
  - Tables
  - Views
  - Stored Procedures
  - Functions
- sys

Object Info | Session

No object selected

classroom*

Limit to 1000 rows

```
14  VALUES
15  (101, "anil", 78, "C", "Pune"),
16  (102, "bhumika", 93, "A", "Mumbai"),
17  (103, "chetan", 85, "B", "Mumbai"),
18  (104, "dhruv", 96, "A", "Delhi"),
19  (105, "emanuel", 12, "F", "Delhi"),
20  (106, "farah", 82, "B", "Delhi");
21
22  SELECT DISTINCT city FROM student;
23
24  SELECT city
25  FROM student
26  WHERE grade = "A"
27  GROUP BY city
28  HAVING MAX(marks) >= 93
29  ORDER BY city DESC;
30
```

160% 4:24

Action Output

	Time	Action	Resp
44	14:06:15	SELECT city FROM student GROUP BY city LIMIT 0, 1000	3 row
45	14:06:28	SELECT city, count(rollno) FROM student GROUP BY city LIMIT 0, 1000	3 row
46	14:07:01	SELECT city, count(rollno) FROM student GROUP BY city HAVING MAX(marks) > 90 LIMIT 0, 1000	2 row
47	14:11:04	SELECT city FROM student WHERE grade = "A" GROUP BY city LIMIT 0, 1000	2 row
48	14:11:46	SELECT city FROM student WHERE grade = "A" GROUP BY city HAVING MAX(marks) > 93 LIMIT 0, 1000	1 row
49	14:11:58	SELECT city FROM student WHERE grade = "A" GROUP BY city HAVING MAX(marks) >= 93 LIMIT 0, 1000	2 row(s)
50	14:12:20	SELECT city FROM student WHERE grade = "A" GROUP BY city HAVING MAX(marks) >= 93 ORDER BY city ASC LIMIT...	2 row(s)

APNA COLLEGE

Table related Queries

Update (to update existing rows)

```
UPDATE table_name
SET col1 = val1, col2 = val2
WHERE condition;
```

```
UPDATE student
SET grade = "O"
WHERE grade = "A";
```

APNA COLLEGE

safe mode

Table related Queries

Update (to update existing rows)

```
UPDATE table_name
SET col1 = val1, col2 = val2
WHERE condition;
```

```
UPDATE student
SET grade = "O"
WHERE grade = "A";
```

SET SQL_SAFE_UPDATES=0;

Table related Queries

Update (to update existing rows)

```
UPDATE table_name
SET col1 = val1, col2 = val2
WHERE condition;
```

```
UPDATE student
SET grade = "O"
WHERE grade = "A";
```

APNA COLLEGE

Administration | Schemas

SCHEMAS

Filter objects

- college
  - Tables
  - Views
  - Stored Procedures
  - Functions
- sys

Object Info | Session

No object selected

classroom*

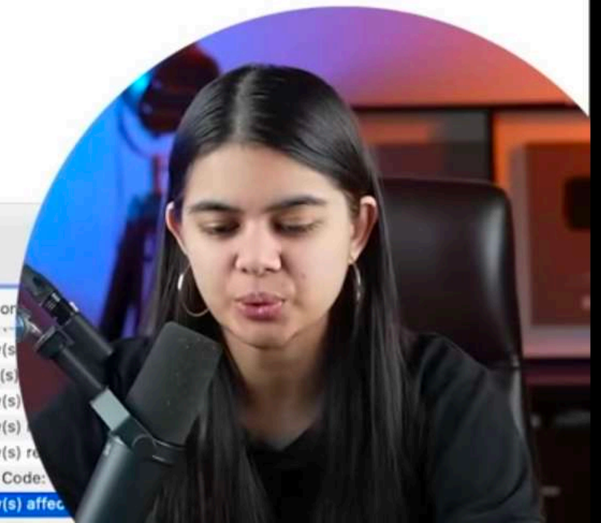
Limit to 1000 rows

```
13  (rollno, name, marks, grade, city)
14  VALUES
15  (101, "anil", 78, "C", "Pune"),
16  (102, "bhumika", 93, "A", "Mumbai"),
17  (103, "chetan", 85, "B", "Mumbai"),
18  (104, "dhruv", 96, "A", "Delhi"),
19  (105, "emanuel", 12, "F", "Delhi"),
20  (106, "farah", 82, "B", "Delhi");
21
22  SET SQL_SAFE_UPDATES = 0;
23
24  UPDATE student
25  SET grade = "O"
26  WHERE grade = "A";
27
28
29
30
```

160% 1:22

Action Output

	Time	Action	Response
47	14:11:04	SELECT city FROM student WHERE grade = "A" GROUP BY city LIMIT 0, 1000	2 row(s)
48	14:11:46	SELECT city FROM student WHERE grade = "A" GROUP BY city HAVING MAX(marks) > 93 LIMIT 0, 1000	1 row(s)
49	14:11:58	SELECT city FROM student WHERE grade = "A" GROUP BY city HAVING MAX(marks) >= 93 LIMIT 0, 1000	2 row(s)
50	14:12:20	SELECT city FROM student WHERE grade = "A" GROUP BY city HAVING MAX(marks) >= 93 ORDER BY city ASC LIMIT...	2 row(s)
51	14:12:28	SELECT city FROM student WHERE grade = "A" GROUP BY city HAVING MAX(marks) >= 93 ORDER BY city DESC LIMI...	2 row(s) r
52	14:28:50	UPDATE student SET grade = "O" WHERE grade = "A"	Error Code:
53	14:30:19	SET SQL_SAFE_UPDATES = 0	0 row(s) affec



APNA COLLEGE

Administration | Schemas

SCHEMAS

Filter objects

- college
  - Tables
  - Views
  - Stored Procedures
  - Functions
- sys

Object Info | Session

No object selected

classroom*

Limit to 1000 rows

```
    VALUES
15  (101, "anil", 78, "C", "Pune"),
16  (102, "bhumika", 93, "A", "Mumbai"),
17  (103, "chetan", 85, "B", "Mumbai"),
18  (104, "dhruv", 96, "A", "Delhi"),
19  (105, "emanuel", 12, "F", "Delhi"),
20  (106, "farah", 82, "B", "Delhi");
21
22  SET SQL_SAFE_UPDATES = 0;
23
24  UPDATE student
25  SET grade = "B"
26  WHERE marks BETWEEN 80 AND 90;
27
28  SELECT * FROM student;
29
30
31
```

160% 15:24

Action Output

	Time	Action	Respon
53	14:30:19	SET SQL_SAFE_UPDATES = 0	0 row(s)
54	14:30:25	UPDATE student SET grade = "O" WHERE grade = "A"	2 row(s)
55	14:30:38	SELECT * FROM student LIMIT 0, 1000	6 row(s)
56	14:31:24	UPDATE student SET marks = 82 WHERE rollno = 105	1 row(s)
57	14:31:27	SELECT * FROM student LIMIT 0, 1000	6 row(s)
58	14:32:10	UPDATE student SET grade = "B" WHERE marks BETWEEN 80 AND 90	1 row(s) aff
59	14:32:13	SELECT * FROM student LIMIT 0, 1000	6 row(s) retur

APNA COLLEGE

Administration | Schemas | classroom*

SCHEMAS

Filter objects

- college
  - Tables
  - Views
  - Stored Procedures
  - Functions
- sys

Object Info | Session

No object selected

Limit to 1000 rows

```
14  VALUES
15  (101, "anil", 78, "C", "Pune"),
16  (102, "bhumika", 93, "A", "Mumbai"),
17  (103, "chetan", 85, "B", "Mumbai"),
18  (104, "dhruv", 96, "A", "Delhi"),
19  (105, "emanuel", 12, "F", "Delhi"),
20  (106, "farah", 82, "B", "Delhi");
21
22  SET SQL_SAFE_UPDATES = 0;
23
24  UPDATE student
25  SET marks = marks + 1;
26
27  SELECT * FROM student;
28
29
30
```

160% 1:24

Action Output

	Time	Action	Response
53	14:30:19	SET SQL_SAFE_UPDATES = 0	0 row(s)
54	14:30:25	UPDATE student SET grade = "O" WHERE grade = "A"	2 row(s)
55	14:30:38	SELECT * FROM student LIMIT 0, 1000	6 row(s)
56	14:31:24	UPDATE student SET marks = 82 WHERE rollno = 105	1 row(s)
57	14:31:27	SELECT * FROM student LIMIT 0, 1000	6 row(s)
58	14:32:10	UPDATE student SET grade = "B" WHERE marks BETWEEN 80 AND 90	1 row(s) aff
59	14:32:13	SELECT * FROM student LIMIT 0, 1000	6 row(s) retur

APNA COLLEGE

Table related Queries

Delete (to delete existing rows)

```
DELETE FROM table_name
WHERE condition;
```

```
DELETE FROM student
WHERE marks < 33;
```

Revisiting FK

Revisiting FK

Foreign Key(dept_id)
REFERENCES
dept

dept

id	name
101	Science
102	English
103	Hindi

FK

teacher

id	name	dept_id (FK)
101	Adam	101
102	Bob	103
103	Casey	102
104	Donald	102

Administration | Schemas | classroom*

SCHEMAS

Filter objects

- college
  - Tables
  - Views
  - Stored Procedures
  - Functions
- sys

Object Info | Session

No object selected

Limit to 1000 rows

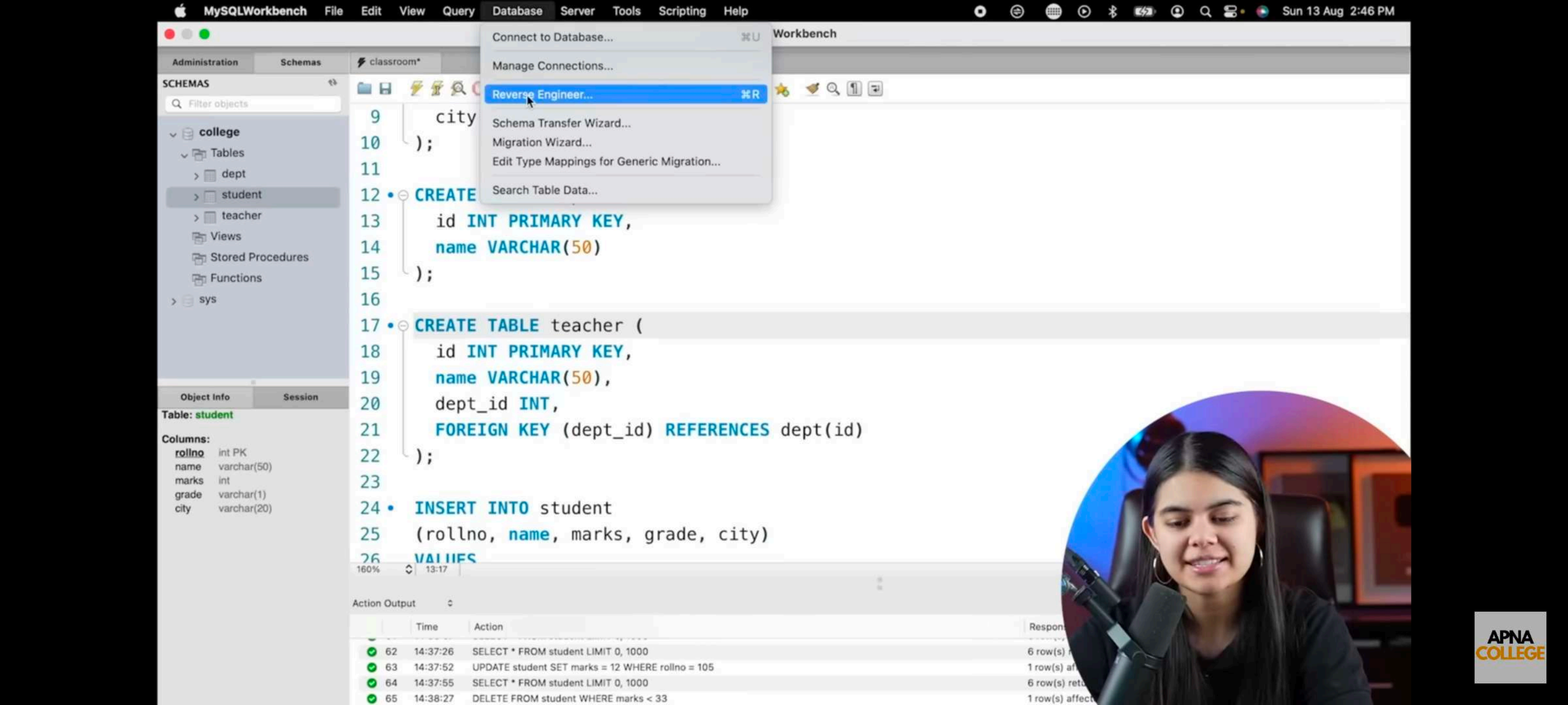
```
9     city VARCHAR(20)
10  );
11
12  CREATE TABLE dept (
13    id INT PRIMARY KEY,
14    name VARCHAR(50)
15  );
16
17  CREATE TABLE teacher (
18    id INT PRIMARY KEY,
19    name VARCHAR(50),
20    dept_id INT,
21    FOREIGN KEY (dept_id) REFERENCES dept(id)
22  );
23
24  INSERT INTO student
25  (rollno, name, marks, grade, city)
26  VALUES
```

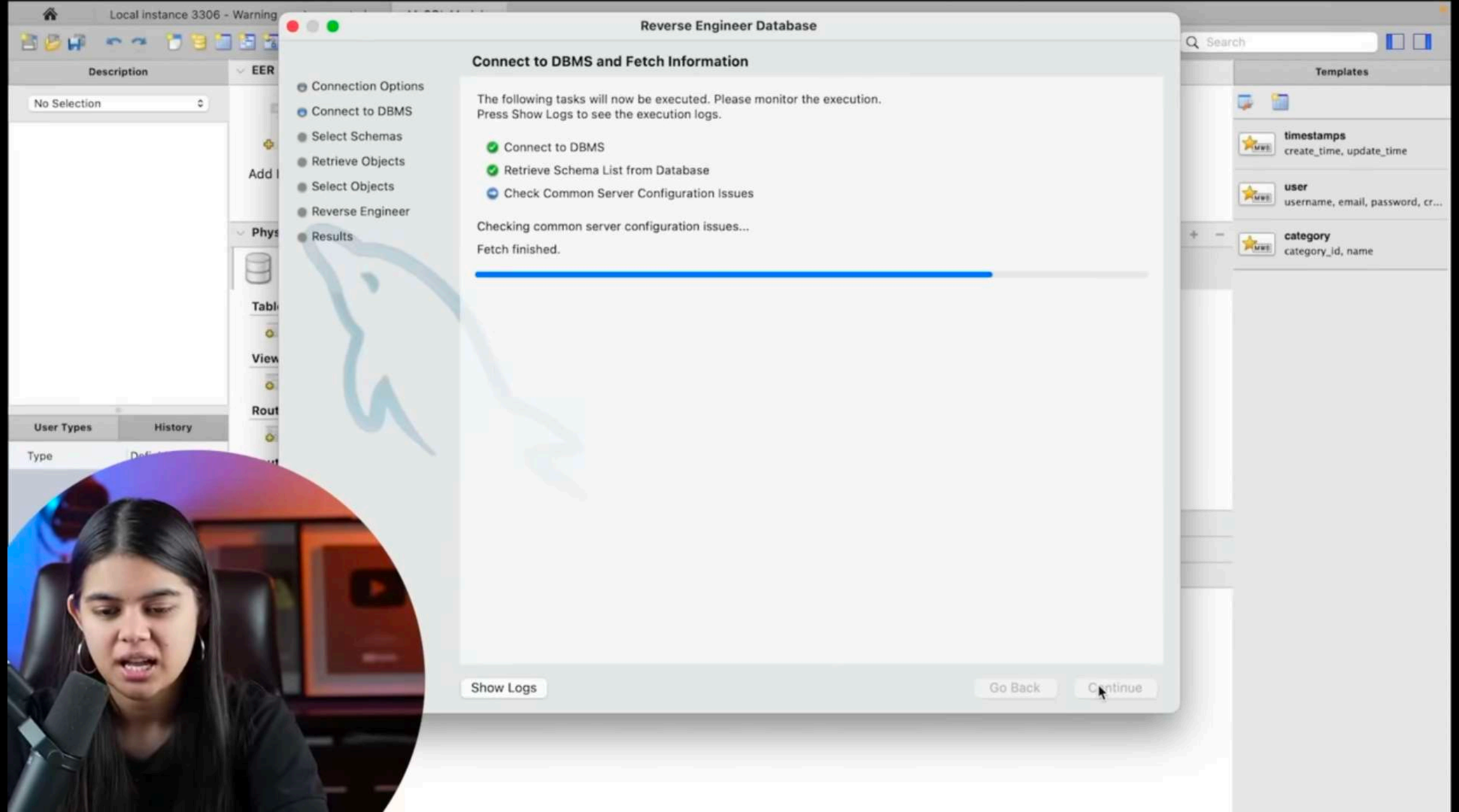
160% 4:19

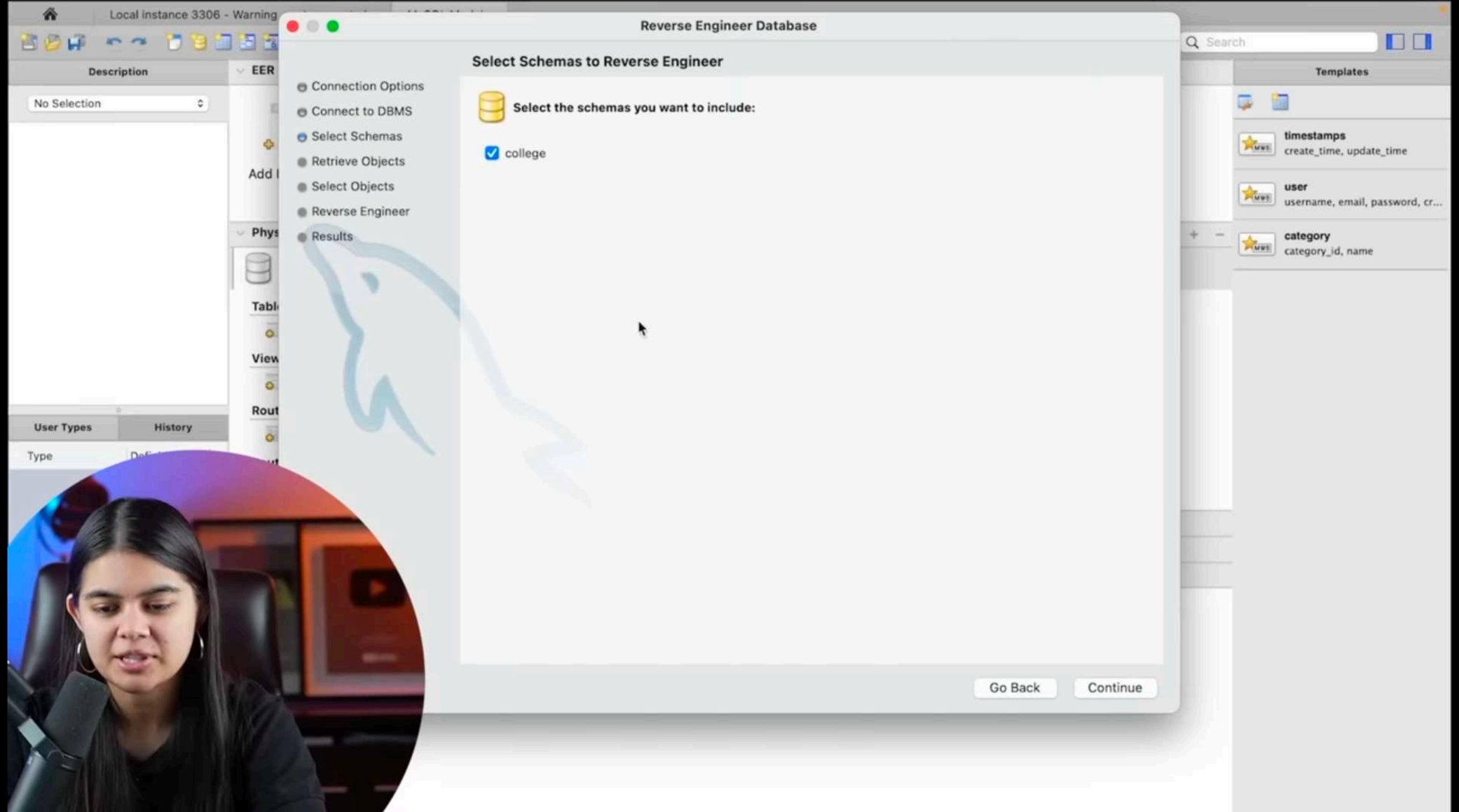
Action Output

	Time	Action	Respon
60	14:33:14	UPDATE student SET marks = marks + 1	6 row(s)
61	14:33:37	SELECT * FROM student LIMIT 0, 1000	6 row(s)
62	14:37:26	SELECT * FROM student LIMIT 0, 1000	6 row(s)
63	14:37:52	UPDATE student SET marks = 12 WHERE rollno = 105	1 row(s)
64	14:37:55	SELECT * FROM student LIMIT 0, 1000	6 row(s)
65	14:38:27	DELETE FROM student WHERE marks < 33	1 row(s)
66	14:38:32	SELECT * FROM student LIMIT 0, 1000	5 row(s)

APNA COLLEGE


MySQLWorkbench File Edit View Query Database Server Tools Scripting Help
Sun 13 Aug 2:46 PM
Workbench
Connect to Database... ⌘U
Manage Connections...
Reverse Engineer... ⌘R
Schema Transfer Wizard...
Migration Wizard...
Edit Type Mappings for Generic Migration...
Search Table Data...
Administration Schemas
classroom*
SCHEMAS
Filter objects
college
Tables
dept
student
teacher
Views
Stored Procedures
Functions
sys
Object Info Session
Table: student
Columns:
rollno int PK
name varchar(50)
marks int
grade varchar(1)
city varchar(20)
city
);
CREATE
id INT PRIMARY KEY,
name VARCHAR(50)
);
CREATE TABLE teacher (
id INT PRIMARY KEY,
name VARCHAR(50),
dept_id INT,
FOREIGN KEY (dept_id) REFERENCES dept(id)
);
INSERT INTO student
(rollno, name, marks, grade, city)
VALUES
160% 13:17
Action Output
Time Action
62 14:37:26 SELECT * FROM student LIMIT 0, 1000
63 14:37:52 UPDATE student SET marks = 12 WHERE rollno = 105
64 14:37:55 SELECT * FROM student LIMIT 0, 1000
65 14:38:27 DELETE FROM student WHERE marks < 33
APNA COLLEGE


Local instance 3306 - Warning
Description
No Selection
EER
Add
Phys
Tabl
View
Rout
User Types
History
Type
Reverse Engineer Database
Connection Options
Connect to DBMS
Select Schemas
Retrieve Objects
Select Objects
Reverse Engineer
Results
Connect to DBMS and Fetch Information
The following tasks will now be executed. Please monitor the execution.
Press Show Logs to see the execution logs.
Connect to DBMS
Retrieve Schema List from Database
Check Common Server Configuration Issues
Checking common server configuration issues...
Fetch finished.
Show Logs
Go Back
Continue
Search
Templates
timestamps
create_time, update_time
user
username, email, password, cr...
category
category_id, name


Local instance 3306 - Warning
Reverse Engineer Database
Search
Description
No Selection
EER
Templates
Connection Options
Connect to DBMS
Select Schemas
Retrieve Objects
Select Objects
Reverse Engineer
Results
Select Schemas to Reverse Engineer
Select the schemas you want to include:
college
timestamps
create_time, update_time
user
username, email, password, cr...
category
category_id, name
User Types
History
Type
Go Back
Continue
APNA COLLEGE

Local instance 3306 - Warning - not supported | MySQL Model* | EER Diagram

Search

Navigator

100

Catalog | Layers | User Types

- mydb
  - Tables
  - Views
  - Routine Groups
- college

Description | Properties | History

No Selection

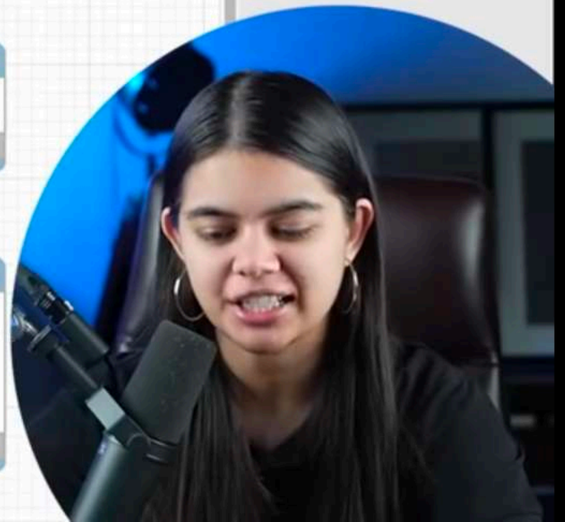
Templates

- timestamps
  create_time, update_time
- user
  username, email, password, cr...
- category
  category_id, name

teacher
- id INT
- name VARCHAR(50)
- dept_id INT
- Indexes

dept
- id INT
- name VARCHAR(50)
- Indexes

student
- rollno INT
- name VARCHAR(50)
- marks INT
- grade VARCHAR(1)
- city VARCHAR(20)
- Indexes



APNA COLLEGE

Revisiting FK

Foreign Key(dept_id)
REFERENCES
dept

dept

id (PK)	name
101	Science
102	English
103	Hindi

FK

teacher

id	name	dept_id (FK)
101	Adam	101
102	Bob	103
103	Casey	102
104	Donald	102

parent table

child table

Cascading for FK

On Update Cascade

When we create a foreign key using this option, it deletes the referencing rows in the child table when the referenced row is deleted in the parent table which has a primary key.

On Delete Cascade

When we create a foreign key using UPDATE CASCADE the referencing rows are updated in the child table when the referenced row is updated in the parent table which has a primary key.

```
CREATE TABLE student (
  id INT PRIMARY KEY,
  courseID INT,
  FOREIGN KEY(courseID) REFERENCES course(id)
  ON DELETE CASCADE
  ON UPDATE CASCADE
);
```

Cascading for FK

On ~~Update~~ Delete Cascade

When we create a foreign key using this option, it deletes the referencing rows in the child table when the referenced row is deleted in the parent table which has a primary key.

On ~~Delete~~ Update Cascade

When we create a foreign key using UPDATE CASCADE the referencing rows are updated in the child table when the referenced row is updated in the parent table which has a primary key.

```sql
CREATE TABLE student (
  id INT PRIMARY KEY,
  courseID INT,
  FOREIGN KEY(courseID) REFERENCES course(id)
  ON DELETE CASCADE
  ON UPDATE CASCADE
);
```

APNA COLLEGE

Administration | Schemas | classroom*

SCHEMAS

Filter objects

- college
  - Tables
    - dept
    - student
    - teacher
  - Views
  - Stored Procedures
  - Functions
- sys

Object Info | Session

Table: student

Columns:

rollno	int PK
name	varchar(50)
marks	int
grade	varchar(1)
city	varchar(20)

Limit to 1000 rows

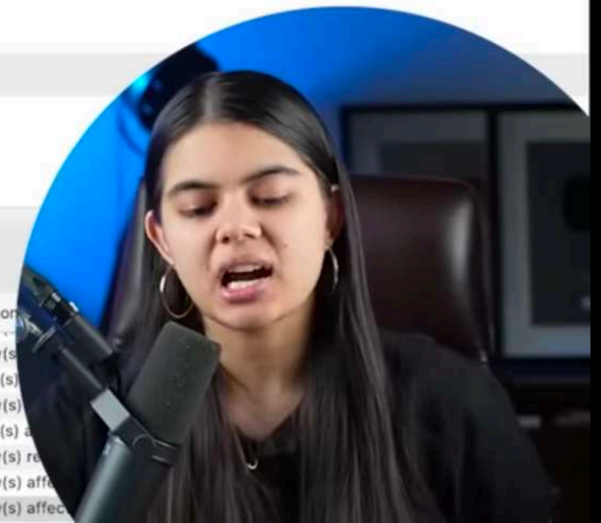
```
9     city VARCHAR(20)
10  );
11
12  CREATE TABLE dept (
13    id INT PRIMARY KEY,
14    name VARCHAR(50)
15  );
16
17  CREATE TABLE teacher (
18    id INT PRIMARY KEY,
19    name VARCHAR(50),
20    dept_id INT,
21    FOREIGN KEY (dept_id) REFERENCES dept(id)
22    ON UPDATE CASCADE
23    ON DELETE CASCADE
24  );
25
26  INSERT INTO student
```

KEYW cas...

160% 19:23

Action Output

	Time	Action	Respon
62	14:37:26	SELECT * FROM student LIMIT 0, 1000	6 row(s)
63	14:37:52	UPDATE student SET marks = 12 WHERE rollno = 105	1 row(s)
64	14:37:55	SELECT * FROM student LIMIT 0, 1000	6 row(s)
65	14:38:27	DELETE FROM student WHERE marks < 33	1 row(s) a
66	14:38:32	SELECT * FROM student LIMIT 0, 1000	5 row(s) re
67	14:46:18	CREATE TABLE dept (id INT PRIMARY KEY, name VARCHAR(50))	0 row(s) affe
68	14:46:27	CREATE TABLE teacher (id INT PRIMARY KEY, name VARCHAR(50), dept_id INT, FOREIGN KEY (dept_id) REFERE...	0 row(s) affec



APNA COLLEGE

Administration | Schemas | classroom*

SCHEMAS

Filter objects

- college
  - Tables
    - dept
    - student
    - teacher
  - Views
  - Stored Procedures
  - Functions
- sys

Object Info | Session

Table: student

Columns:

rollno	int PK
name	varchar(50)
marks	int
grade	varchar(1)
city	varchar(20)

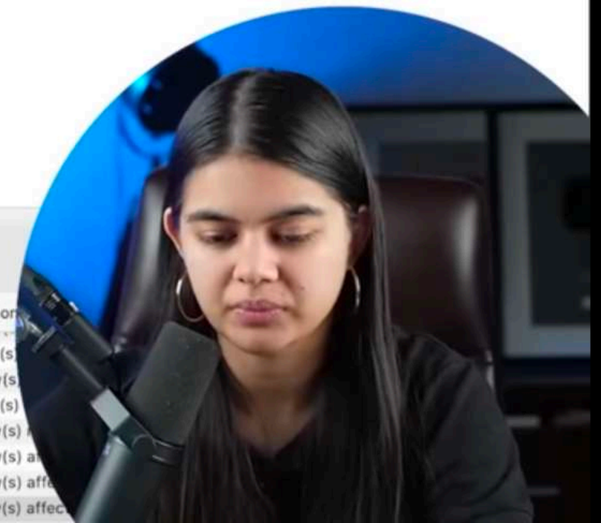
Limit to 1000 rows

```
11
12  CREATE TABLE dept (
13    id INT PRIMARY KEY,
14    name VARCHAR(50)
15  );
16
17  INSERT INTO dept
18  VALUES
19  (101, "english"),
20  (102, "IT");
21
22  SELECT * FROM dept;
23
24  CREATE TABLE teacher (
25    id INT PRIMARY KEY,
26    name VARCHAR(50),
27    dept_id INT,
28    FOREIGN KEY (dept_id) REFERENCES dept(id)
```

160% 20:22

Action Output

	Time	Action	Respon
63	14:37:52	UPDATE student SET marks = 12 WHERE rollno = 105	1 row(s)
64	14:37:55	SELECT * FROM student LIMIT 0, 1000	6 row(s)
65	14:38:27	DELETE FROM student WHERE marks < 33	1 row(s)
66	14:38:32	SELECT * FROM student LIMIT 0, 1000	5 row(s)
67	14:46:18	CREATE TABLE dept (id INT PRIMARY KEY, name VARCHAR(50))	0 row(s) a
68	14:46:27	CREATE TABLE teacher (id INT PRIMARY KEY, name VARCHAR(50), dept_id INT, FOREIGN KEY (dept_id) REFERE...	0 row(s) aff
69	14:57:04	INSERT INTO dept VALUES (101, "english"), (102, "IT")	2 row(s) affec



APNA COLLEGE

Administration | Schemas | classroom*

SCHEMAS

Filter objects

- college
  - Tables
    - dept
    - student
    - teacher
  - Views
  - Stored Procedures
  - Functions
- sys

Object Info | Session

Table: student

Columns:

rollno	int PK
name	varchar(50)
marks	int
grade	varchar(1)
city	varchar(20)

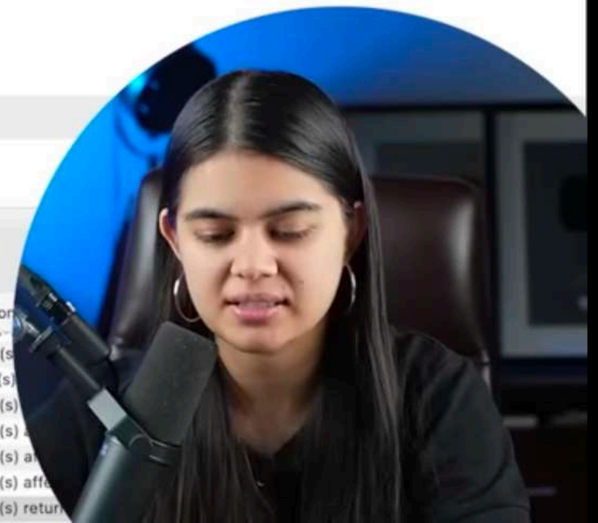
Limit to 1000 rows

```
21
22 • SELECT * FROM dept;
23
24 • CREATE TABLE teacher (
25     id INT PRIMARY KEY,
26     name VARCHAR(50),
27     dept_id INT,
28     FOREIGN KEY (dept_id) REFERENCES dept(id)
29     ON UPDATE CASCADE
30     ON DELETE CASCADE
31 );
32
33 • INSERT INTO teacher
34   VALUES
35   (101, "Adam", 101),
36   (102, "Eve", 102);
37
38   INSERT INTO student
```

160% | 19:36

Action Output

	Time	Action	Respon
64	14:37:55	SELECT * FROM student LIMIT 0, 1000	6 row(s)
65	14:38:27	DELETE FROM student WHERE marks < 33	1 row(s)
66	14:38:32	SELECT * FROM student LIMIT 0, 1000	5 row(s)
67	14:46:18	CREATE TABLE dept (id INT PRIMARY KEY, name VARCHAR(50))	0 row(s)
68	14:46:27	CREATE TABLE teacher (id INT PRIMARY KEY, name VARCHAR(50), dept_id INT, FOREIGN KEY (dept_id) REFERE...	0 row(s)
69	14:57:04	INSERT INTO dept VALUES (101, "english"), (102, "IT")	2 row(s) aff
70	14:57:17	SELECT * FROM dept LIMIT 0, 1000	2 row(s) retur



APNA COLLEGE

Administration | Schemas | classroom*

SCHEMAS

Filter objects

- college
  - Tables
    - dept
    - student
    - teacher
  - Views
  - Stored Procedures
  - Functions
- sys

Object Info | Session

Table: student

Columns:

rollno	int PK
name	varchar(50)
marks	int
grade	varchar(1)
city	varchar(20)

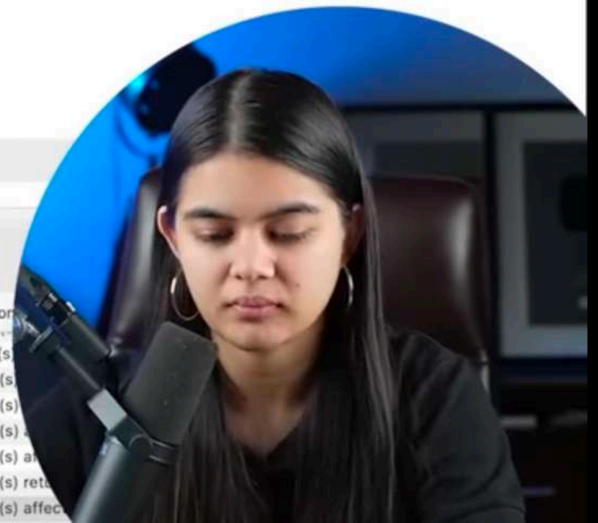
Limit to 1000 rows

```
22 • SELECT * FROM dept;
23
24 • CREATE TABLE teacher (
25     id INT PRIMARY KEY,
26     name VARCHAR(50),
27     dept_id INT,
28     FOREIGN KEY (dept_id) REFERENCES dept(id)
29     ON UPDATE CASCADE
30     ON DELETE CASCADE
31 );
32
33 • INSERT INTO teacher
34   VALUES
35   (101, "Adam", 101),
36   (102, "Eve", 102);
37
38 • SELECT * FROM teacher;
```

160% 2:38

Action Output

	Time	Action	Respon
65	14:38:27	DELETE FROM student WHERE marks < 33	1 row(s)
66	14:38:32	SELECT * FROM student LIMIT 0, 1000	5 row(s)
67	14:46:18	CREATE TABLE dept (id INT PRIMARY KEY, name VARCHAR(50))	0 row(s)
68	14:46:27	CREATE TABLE teacher (id INT PRIMARY KEY, name VARCHAR(50), dept_id INT, FOREIGN KEY (dept_id) REFERE...	0 row(s)
69	14:57:04	INSERT INTO dept VALUES (101, "english"), (102, "IT")	2 row(s) a
70	14:57:17	SELECT * FROM dept LIMIT 0, 1000	2 row(s) ret
71	14:57:57	INSERT INTO teacher VALUES (101, "Adam", 101), (102, "Eve", 102)	2 row(s) affec



APNA COLLEGE

Administration | Schemas

SCHEMAS

Filter objects

- college
  - Tables
    - dept
    - student
  - Views
  - Stored Procedures
  - Functions
- sys

Object Info | Session

Table: student

Columns:

rollno	int PK
name	varchar(50)
marks	int
grade	varchar(1)
city	varchar(20)

classroom*

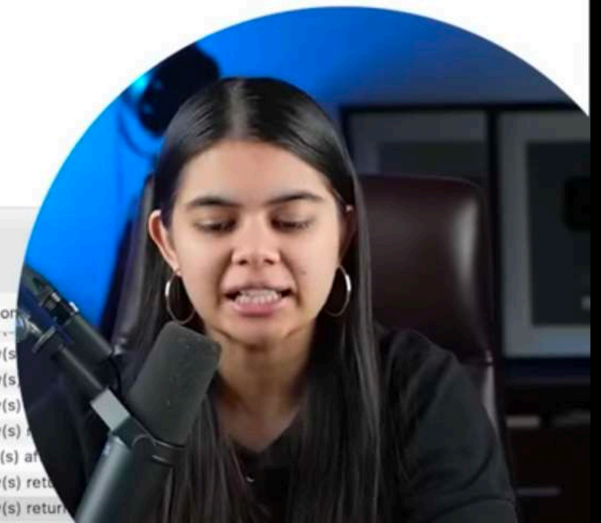
Limit to 1000 rows

```
INSERT INTO dept
VALUES
(101, "english"),
(102, "IT");

SELECT * FROM dept;

UPDATE dept
SET id = 103
WHERE id = 102;

CREATE TABLE teacher (
  id INT PRIMARY KEY,
  name VARCHAR(50),
  dept_id INT,
  FOREIGN KEY (dept_id) REFERENCES dept(id)
  ON UPDATE CASCADE
```

160% 15:19

Action Output

	Time	Action	Respon
73	14:58:40	DROP TABLE teacher	0 row(s)
74	14:58:46	CREATE TABLE teacher (id INT PRIMARY KEY, name VARCHAR(50), dept_id INT, FOREIGN KEY (dept_id) REFERE...	0 row(s)
75	14:58:50	INSERT INTO teacher VALUES (101, "Adam", 101), (102, "Eve", 102)	2 row(s)
76	14:58:53	SELECT * FROM teacher LIMIT 0, 1000	2 row(s)
77	14:59:29	UPDATE dept SET id = 103 WHERE id = 102	1 row(s) af
78	14:59:34	SELECT * FROM dept LIMIT 0, 1000	2 row(s) ret
79	14:59:43	SELECT * FROM teacher LIMIT 0, 1000	2 row(s) retur



APNA COLLEGE

Table related Queries

Alter (to change the schema)

ADD Column

```
ALTER TABLE table_name
ADD COLUMN column_name datatype constraint;
```

DROP Column

```
ALTER TABLE table_name
DROP COLUMN column_name;
```

RENAME Table

```
ALTER TABLE table_name
RENAME TO new_table_name;
```

Administration | Schemas

SCHEMAS

Filter objects

- college
  - Tables
    - dept
    - student
  - Views
  - Stored Procedures
  - Functions
- sys

Object Info | Session

Table: dept

Columns:

id	int PK
name	varchar(50)

classroom*

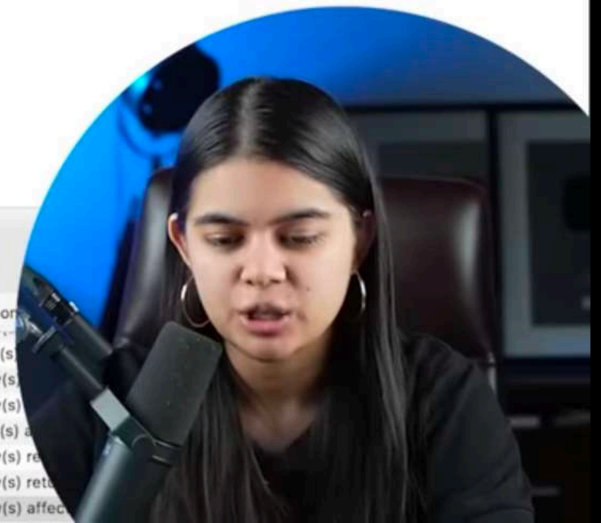
Limit to 1000 rows

```
13  (rollno, name, marks, grade, city)
14  VALUES
15  (101, "anil", 78, "C", "Pune"),
16  (102, "bhumika", 93, "A", "Mumbai"),
17  (103, "chetan", 85, "B", "Mumbai"),
18  (104, "dhruv", 96, "A", "Delhi"),
19  (105, "emanuel", 12, "F", "Delhi"),
20  (106, "farah", 82, "B", "Delhi");
21
22  SELECT * FROM student;
23
24  ALTER TABLE student
25  ADD COLUMN age INT;
26
27
28
29
30
```

160% 1:22

Action Output

	Time	Action	Respon
77	14:59:29	UPDATE dept SET id = 103 WHERE id = 102	1 row(s)
78	14:59:34	SELECT * FROM dept LIMIT 0, 1000	2 row(s)
79	14:59:43	SELECT * FROM teacher LIMIT 0, 1000	2 row(s)
80	15:00:09	UPDATE dept SET id = 111 WHERE id = 101	1 row(s)
81	15:00:13	SELECT * FROM dept LIMIT 0, 1000	2 row(s)
82	15:00:24	SELECT * FROM teacher LIMIT 0, 1000	2 row(s)
83	15:03:30	ALTER TABLE student ADD COLUMN age INT	0 row(s)



APNA COLLEGE

Administration | Schemas

SCHEMAS

Filter objects

- college
  - Tables
    - dept
    - student
  - Views
  - Stored Procedures
  - Functions
- sys

Object Info | Session

Table: dept

Columns:

id	int PK
name	varchar(50)

classroom*

Limit to 1000 rows

```
14  VALUES
15  (101, "anil", 78, "C", "Pune"),
16  (102, "bhumika", 93, "A", "Mumbai"),
17  (103, "chetan", 85, "B", "Mumbai"),
18  (104, "dhruv", 96, "A", "Delhi"),
19  (105, "emanuel", 12, "F", "Delhi"),
20  (106, "farah", 82, "B", "Delhi");
21
22  SELECT * FROM student;
23
24  ALTER TABLE student
25  DROP COLUMN age;
26
27
28
29
30
```

160% 1:24

Action Output

	Time	Action	Respon
78	14:59:34	SELECT * FROM dept LIMIT 0, 1000	2 row(s
79	14:59:43	SELECT * FROM teacher LIMIT 0, 1000	2 row(s
80	15:00:09	UPDATE dept SET id = 111 WHERE id = 101	1 row(s)
81	15:00:13	SELECT * FROM dept LIMIT 0, 1000	2 row(s)
82	15:00:24	SELECT * FROM teacher LIMIT 0, 1000	2 row(s) re
83	15:03:30	ALTER TABLE student ADD COLUMN age INT	0 row(s) aff
84	15:03:33	SELECT * FROM student LIMIT 0, 1000	5 row(s) retur

APNA COLLEGE

Table related Queries

CHANGE Column (rename)

```
ALTER TABLE table_name
CHANGE COLUMN old_name new_name new_datatype new_constraint;
```

MODIFY Column (modify datatype/ constraint)

```
ALTER TABLE table_name
MODIFY col_name new_datatype new_constraint;
```

ADD Column

```
ALTER TABLE student
ADD COLUMN age INT NOT NULL DEFAULT 19;
```

DROP Column

```
ALTER TABLE student
DROP COLUMN stu_age;
```

MODIFY Column

```
ALTER TABLE student
MODIFY age VARCHAR(2);
```

RENAME Table

```
ALTER TABLE student
RENAME TO stu;
```

CHANGE Column (rename)

```
ALTER TABLE student
CHANGE age stu_age INT;
```

APNA COLLEGE

Table related Queries

Truncate (to delete table's data)

```
TRUNCATE TABLE table_name ;
```

```
UPDATE student
SET grade = "0"
WHERE grade = "A";
```

APNA COLLEGE

Table related Queries

Truncate (to delete table's data)

```
TRUNCATE TABLE table_name ;
```

DROP → delete table.

TRUNCATE → table data

```
UPDATE student
SET grade = "O"
WHERE grade = "A";
```

APNA COLLEGE

Administration | Schemas | classroom*

SCHEMAS

Filter objects

- college
  - Tables
    - dept
    - student
    - teacher
  - Views
  - Stored Procedures
  - Functions
- sys

Object Info | Session

Table: student

Columns:

rollno	int PK
name	varchar(50)
marks	int
grade	varchar(1)
city	varchar(20)

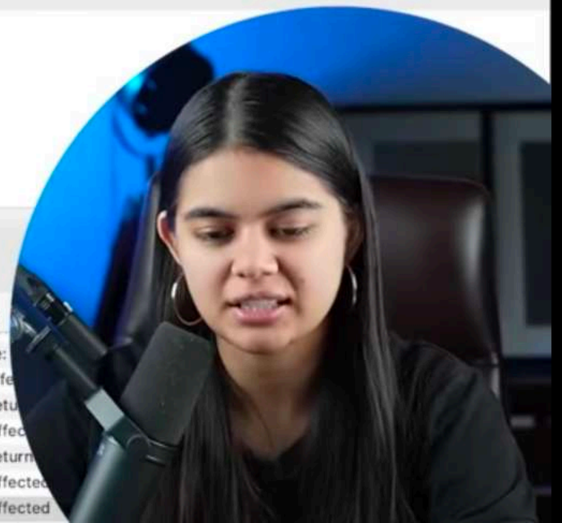
Limit to 1000 rows

```
12 • INSERT INTO student
13   (rollno, name, marks, grade, city)
14   VALUES
15   (101, "anil", 78, "C", "Pune"),
16   (102, "bhumika", 93, "A", "Mumbai"),
17   (103, "chetan", 85, "B", "Mumbai"),
18   (104, "dhruv", 96, "A", "Delhi"),
19   (105, "emanuel", 12, "F", "Delhi"),
20   (106, "farah", 82, "B", "Delhi");
21
22 • SELECT * FROM student;
23
24 • TRUNCATE TABLE student;
25
26
27
28
29
```

160% 1:24

Action Output

	Time	Action	Response
94	15:09:31	INSERT INTO student (rollno, name, marks, age) VALUES (107, "gargi", 68, 100)	Error Code:
95	15:09:39	INSERT INTO student (rollno, name, marks, stu_age) VALUES (107, "gargi", 68, 100)	1 row(s) affe
96	15:09:43	SELECT * FROM student LIMIT 0, 1000	6 row(s) retu
97	15:10:14	ALTER TABLE student DROP COLUMN stu_age	0 row(s) affec
98	15:10:18	SELECT * FROM student LIMIT 0, 1000	6 row(s) return
99	15:10:44	ALTER TABLE student RENAME TO stu	0 row(s) affected
100	15:11:01	ALTER TABLE stu RENAME TO student	0 row(s) affected



APNA COLLEGE

Joins in SQL

Join is used to combine rows from two or more tables, based on a related column between them.

Joins in SQL

Join is used to combine rows from two or more tables, based on a related column between them.

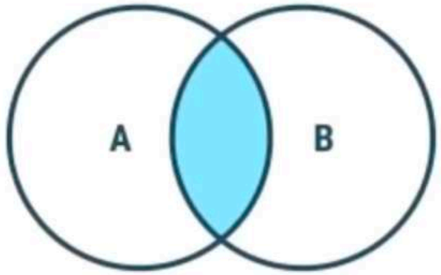
employee

id	name
101	
102	

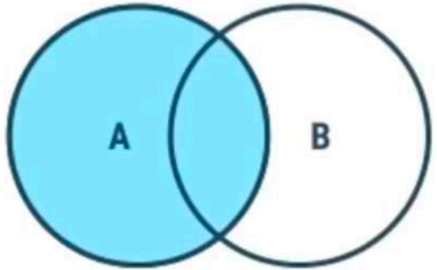
salary

id	salary
102	
103	

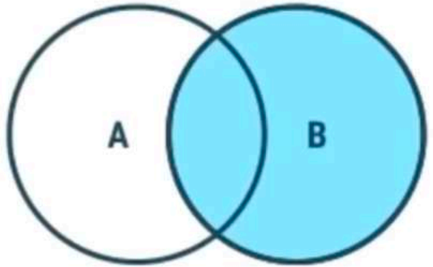
Types of Joins



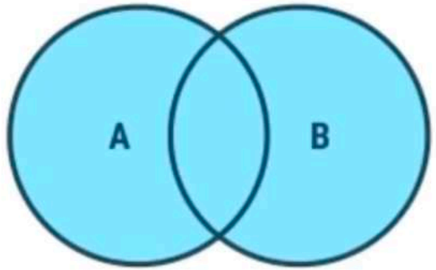
Inner Join



Left Join



Right Join

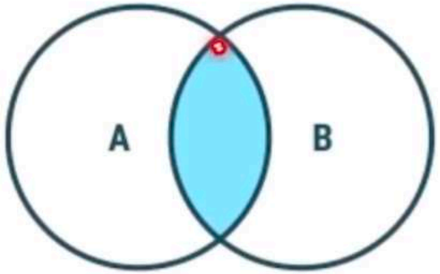


Full Join

Outer Joins

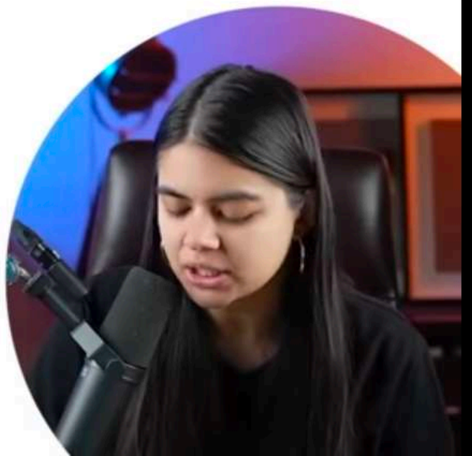
Inner Join

Returns records that have matching values in both tables



Syntax

```
SELECT column(s)
FROM tableA
INNER JOIN tableB
ON tableA.col_name = tableB.col_name;
```



Inner Join

Example

student

student_id	name
101	adam
102	bob
103	casey

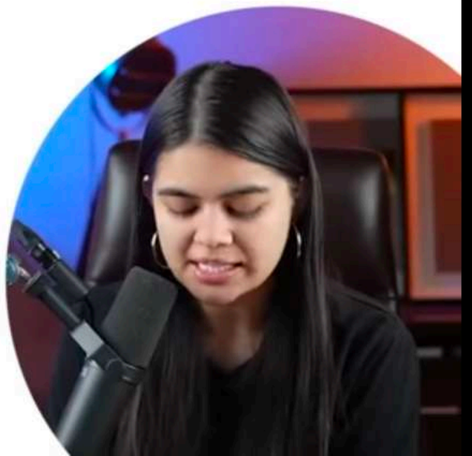
course

student_id	course
102	english
105	math
103	science
107	computer science

```
SELECT *
FROM student
INNER JOIN course
ON student.student_id = course.student_id;
```

Result

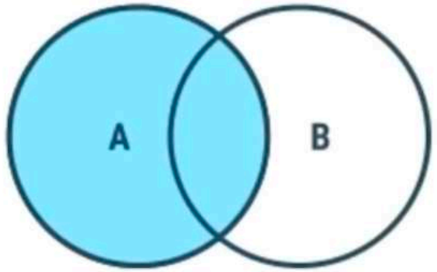
student_id	name	course
102	bob	english
103	casey	science



APNA COLLEGE

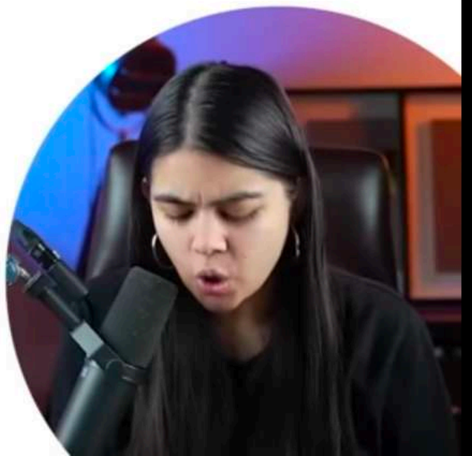
Left Join

Returns all records from the left table, and the matched records from the right table



Syntax

```
SELECT column(s)
FROM tableA
LEFT JOIN tableB
ON tableA.col_name = tableB.col_name;
```



APNA COLLEGE

Left Join

Example

student (left)

student_id	name
101	adam
102	bob
103	casey

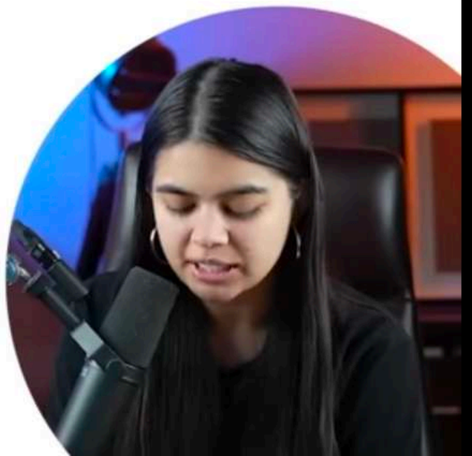
course (Right)

student_id	course
102	english
105	math
103	science
107	computer science

```
SELECT *
FROM student as s
LEFT JOIN course as c
ON s.student_id = c.student_id;
```

Result

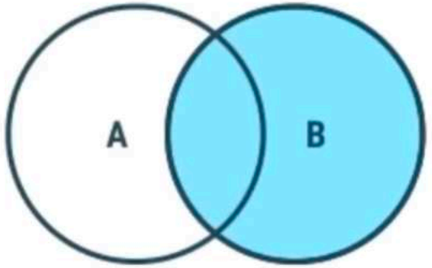
student_id	name	course
101	adam	*null*
102	bob	english
103	casey	science



APNA COLLEGE

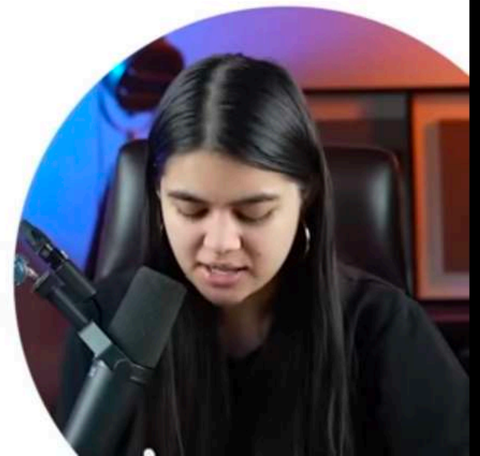
Right Join

Returns all records from the right table, and the matched records from the left table



Syntax

```
SELECT column(s)
FROM tableA
RIGHT JOIN tableB
ON tableA.col_name = tableB.col_name;
```



APNA COLLEGE

Right Join

Example

student

student_id	name
101	adam
102	bob
103	casey

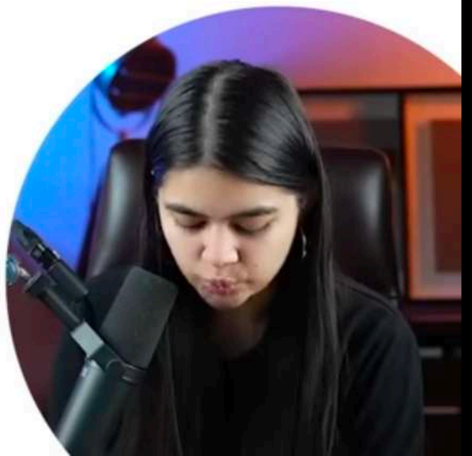
course

student_id	course
102	english
105	math
103	science
107	computer science

Result

student_id	course	name
102	english	bob
105	math	*null*
103	science	casey
107	computer science	*null*

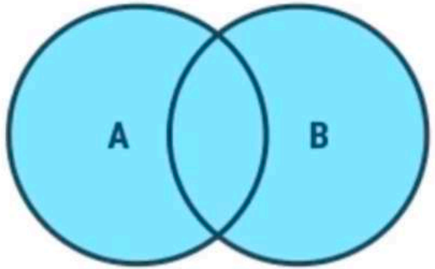
```
SELECT *
FROM student as s
RIGHT JOIN course as c
ON s.student_id = c.student_id;
```



APNA COLLEGE

Full Join

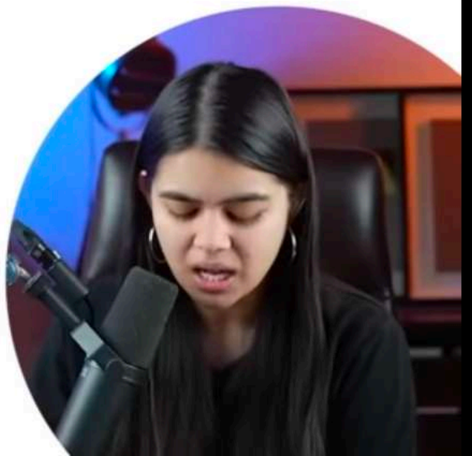
Returns all records when there is a match in either left or right table



Syntax in MySQL

```
SELECT * FROM student as a
LEFT JOIN course as b
ON a.id = b.id
UNION
SELECT * FROM student as a
RIGHT JOIN course as b
ON a.id = b.id;
```

LEFT JOIN
UNION
RIGHT JOIN



APNA COLLEGE

Full Join

Example

student

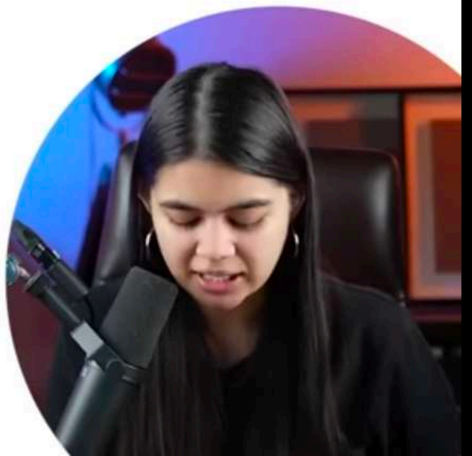
student_id	name
101	adam
102	bob
103	casey

course

student_id	course
102	english
105	math
103	science
107	computer science

Result

student_id	name	course
101	adam	*null*
102	bob	english
103	casey	science
105	*null*	math
107	*null*	computer science



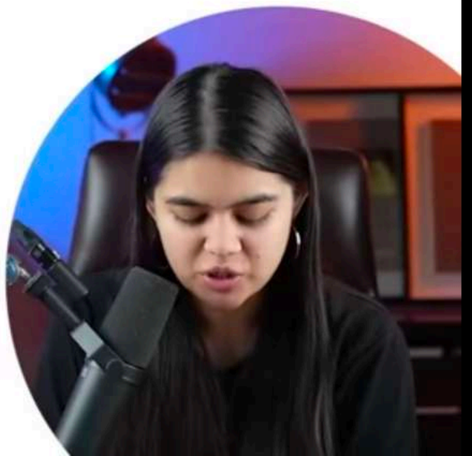
APNA COLLEGE

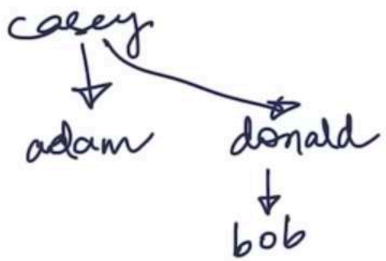
Self Join

It is a regular join but the table is joined with itself.

Syntax

```
SELECT column(s)
FROM table as a
JOIN table as b
ON a.col_name = b.col_name;
```





Self Join

Example

Employee

id	name	manager_id
101	adam	103 (casey)
102	bob	104 (donald)
103	casey	*null*
104	donald	103 (casey)

```
SELECT b.name as manager_name, a.name
FROM employee as a
JOIN employee as b
ON a.id = b.manager_id;
```

a

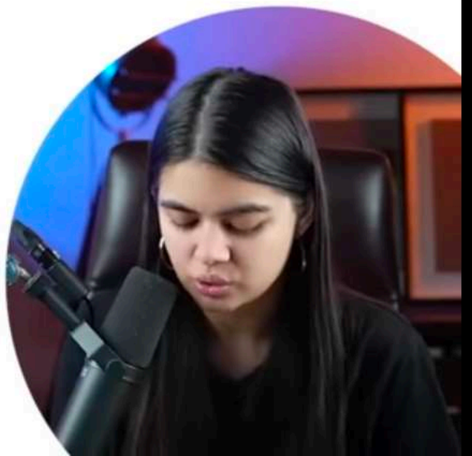
id	name	m-id

b

id	name	m-id

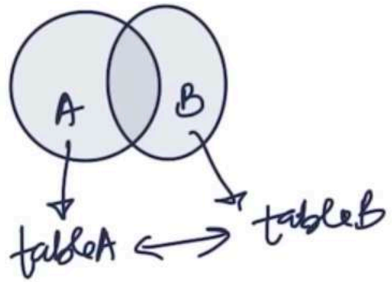
Result

manager_name	name
adam	casey
bob	donald
donald	casey



APNA COLLEGE

Union



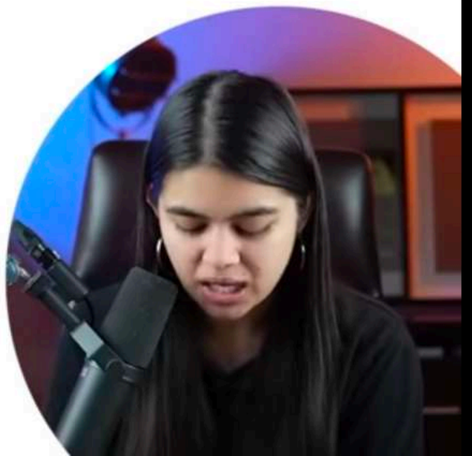
It is used to combine the result-set of two or more SELECT statements.
Gives UNIQUE records.

To use it :

- every SELECT should have same no. of columns
- columns must have similar data types
- columns in every SELECT should be in same order

Syntax

SELECT *column(s)* FROM ***tableA***
UNION
SELECT *column(s)* FROM **tableB**



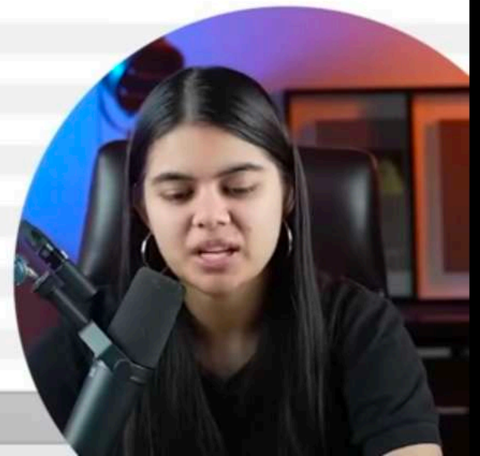
```sql
39
40 • SELECT * FROM employee;
41
42
43 • SELECT name FROM employee
44   UNION
45   SELECT name FROM employee;
```

130% 27:45

Result Grid | Filter Rows: Search | Export:

name
adam
bob
casey
donald

Result 89

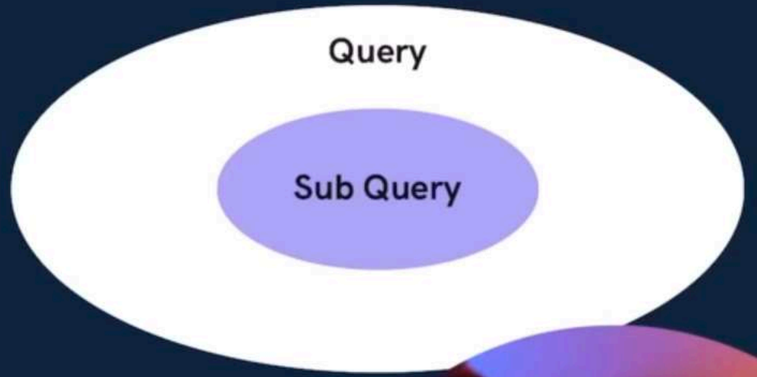


APNA COLLEGE

SQL Sub Queries

A Subquery or Inner query or a Nested query is a query within another SQL query.

It involves 2 select statements.



Syntax

```
SELECT column(s)
FROM table_name
WHERE col_name operator
( subquery );
```

SQL Sub Queries

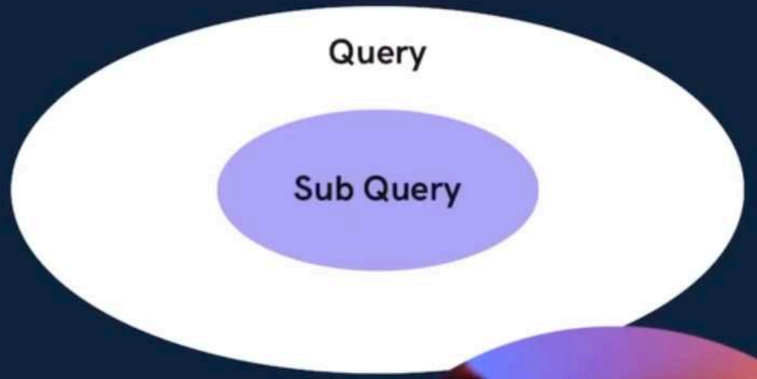
A Subquery or Inner query or a Nested query is a query within another SQL query.

It involves 2 select statements.

Syntax

```
SELECT column(s)
FROM table_name
WHERE col_name operator
( subquery );
```

1) Select
2) From
3) Where



APNA COLLEGE

SQL Sub Queries

Example

Get names of all students who scored more than class average.

Step 1. Find the avg of class
Step 2. Find the names of students with marks > avg

rollno	name	marks
101	anil	78
102	bhumika	93
103	chetan	85
104	dhruv	96
105	emanuel	92
106	farah	82

avg(marks) x

Students marks > avg(marks)

APNA COLLEGE

Administration | Schemas | classroom*

SCHEMAS

Filter objects

- college
  - Tables
  - Views
  - Stored Procedures
  - Functions
- sys

Object Info | Session

No object selected

Limit to 1000 rows

```
CREATE DATABASE college;
USE college;

CREATE TABLE student (
  rollno INT PRIMARY KEY,
  name VARCHAR(50),
  marks INT NOT NULL,
  grade VARCHAR(1),
  city VARCHAR(20)
);

INSERT INTO student
(rollno, name, marks, grade, city)
VALUES
(101, "anil", 78, "C", "Pune"),
(102, "bhumika", 93, "A", "Mumbai"),
(103, "chetan", 85, "B", "Mumbai"),
(104, "dhruv", 96, "A", "Delhi"),
(105, "emanuel", 92, "F", "Delhi"),
(106, "farah", 82, "B", "Delhi");
```

130% 9:9

Action Output

	Time	Action	Respor
140	16:36:54	DROP table student	0 row(s
141	16:36:56	DROP table employee	0 row(s
142	16:37:04	DROP table course	0 row(s
143	17:00:16	CREATE TABLE student (rollno INT PRIMARY KEY, name VARCHAR(50), marks INT NOT NULL, grade VARCHAR(1)...	0 row(s)
144	17:00:20	INSERT INTO student (rollno, name, marks, grade, city) VALUES (101, "anil", 78, "C", "Pune"), (102, "bhumika", 93, "A",...	6 row(s) a
145	17:03:46	UPDATE TABLE student SET marks = 92 WHERE rollno = 105	Error Code:
146	17:03:59	UPDATE student SET marks = 92 WHERE rollno = 105	1 row(s) affec

APNA COLLEGE