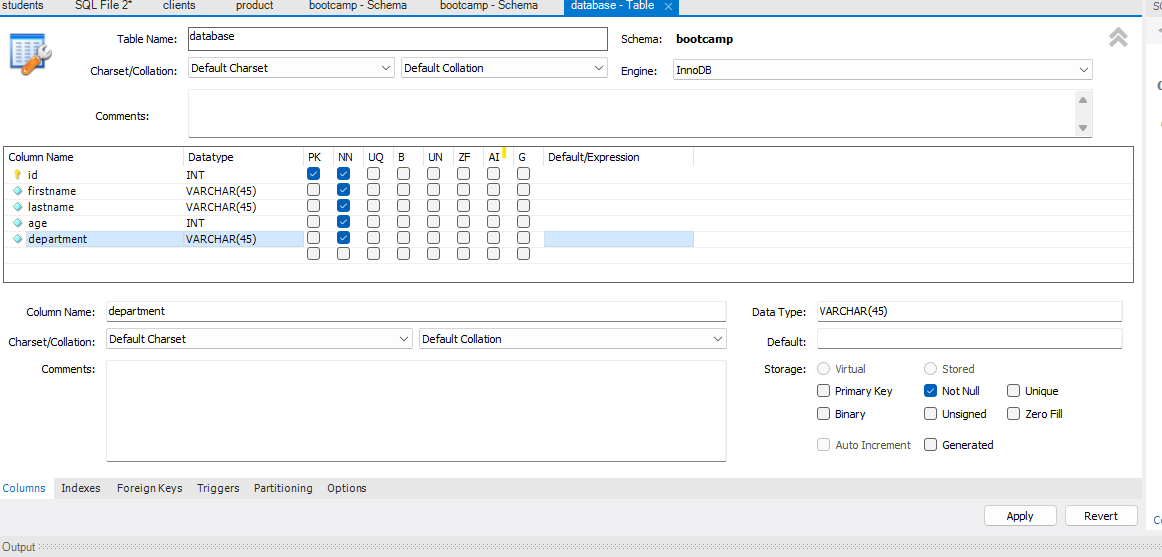
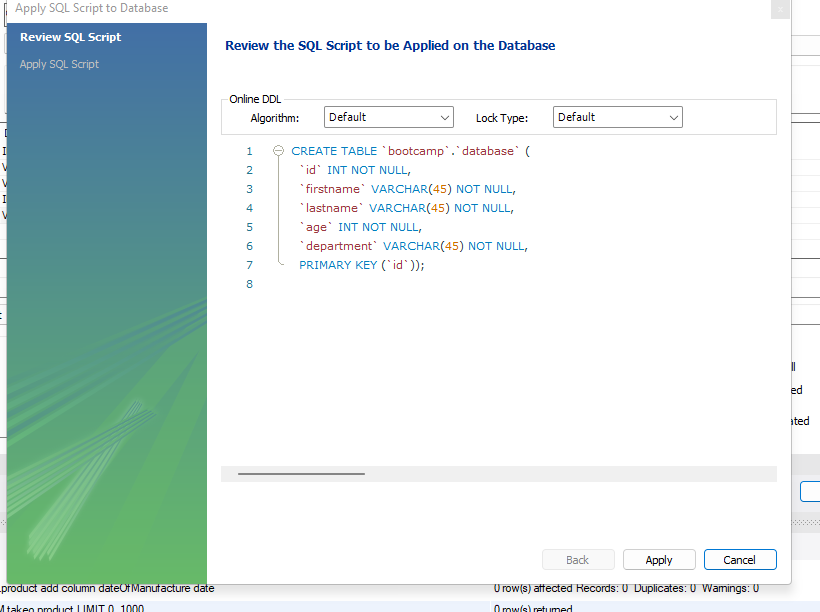


1. Created bootcamp schema.
2. Created a table named database and added columns id, firstname, lastname, age and department





1. Added 15 entries

INSERT INTO `bootcamp`.`database` (`id`, `firstname`, `lastname`, `age`, `department`) VALUES ('100', 'Sita', 'Raman', '36', 'EEE');

INSERT INTO `bootcamp`.`database` (`id`, `firstname`, `lastname`, `age`, `department`) VALUES ('101', 'Phoonsook', 'Wangdu', '57', 'Mech');

INSERT INTO `bootcamp`.`database` (`id`, `firstname`, `lastname`, `age`, `department`) VALUES ('102', 'Jeewan', 'Gyawali', '61', 'CSE');

INSERT INTO `bootcamp`.`database` (`id`, `firstname`, `lastname`, `age`, `department`) VALUES ('103', 'Grishma', 'Parajuli', '21', 'CSE');

INSERT INTO `bootcamp`.`database` (`id`, `firstname`, `lastname`, `age`, `department`) VALUES ('104', 'Hari', 'Sapkota', '25', 'Civil');

INSERT INTO `bootcamp`.`database` (`id`, `firstname`, `lastname`, `age`, `department`) VALUES ('105', 'Krishna', 'KC', '27', 'Chem');

INSERT INTO `bootcamp`.`database` (`id`, `firstname`, `lastname`, `age`, `department`) VALUES ('106', 'Rupak', 'Pandey', '32', 'Civil');

INSERT INTO `bootcamp`.`database` (`id`, `firstname`, `lastname`, `age`, `department`) VALUES ('107', 'Sonu', 'Kumar', '33', 'BioTech');

INSERT INTO `bootcamp`.`database` (`id`, `firstname`, `lastname`, `age`, `department`) VALUES ('108', 'Geetanjali', 'Yadav', '27', 'BioTech');

INSERT INTO `bootcamp`.`database` (`id`, `firstname`, `lastname`, `age`, `department`) VALUES ('109', 'Seema', 'Ganguly', '22', 'ECE');

INSERT INTO `bootcamp`.`database` (`id`, `firstname`, `lastname`, `age`, `department`) VALUES ('110', 'Kalpana', 'Kumari', '23', 'Mech');

INSERT INTO `bootcamp`.`database` (`id`, `firstname`, `lastname`, `age`, `department`) VALUES ('111', 'Debajyoti', 'Khawas', '27', 'Civil');

INSERT INTO `bootcamp`.`database` (`id`, `firstname`, `lastname`, `age`, `department`) VALUES ('112', 'Manu', 'Menon', '29', 'Mech');

INSERT INTO `bootcamp`.`database` (`id`, `firstname`, `lastname`, `age`, `department`) VALUES ('113', 'Niraj', 'Thapa', '33', 'Civil');

INSERT INTO `bootcamp`.`database` (`id`, `firstname`, `lastname`, `age`, `department`) VALUES ('114', 'Prasad', 'Yadav', '33', 'CSE');

INSERT INTO `bootcamp`.`database` (`id`, `firstname`, `lastname`, `age`, `department`) VALUES ('115', 'Kumar', 'Kartikeya', '27', 'Chem');

1. Added one email column :

SELECT \* FROM bootcamp.database;

alter table bootcamp.database add column email varchar(45);

SELECT \* FROM bootcamp.database;

1. Added address column also:

alter table bootcamp.database add column address varchar(55);

SELECT \* FROM bootcamp.database;

1. Dropped address column:

alter table bootcamp.database drop column address;

SELECT \* FROM bootcamp.database;

1. Updated first name and last name for 5 records:

UPDATE bootcamp.database set firstname="sushila", lastname="karki" WHERE id=100;

UPDATE bootcamp.database set firstname="hrishita", lastname="regmi" WHERE id=101;

UPDATE bootcamp.database set firstname="Pritam", lastname="Bhattacharya" WHERE id=102;

UPDATE bootcamp.database set firstname="Krishna", lastname="Kumar" WHERE id=103;

UPDATE bootcamp.database set firstname="Arjun", lastname="Gautam" WHERE id=104;

SELECT \* FROM bootcamp.database;

1. Deleted 4 records where age is less than 18

SELECT \* FROM bootcamp.database;

delete from bootcamp.database where age<18;

SELECT \* FROM bootcamp.database;

SET SQL\_SAFE\_UPDATES = 0;

1. Using aggregate function to the age column:

select avg(age) from bootcamp.database;

Output : avg(age)

28.1667

select count(\*) from bootcamp.database;

Output: count(\*)

12

select count(\*) as Agemorethan30 from bootcamp.database where age>30;

Output Agemorethan30

4

select MAX(age) from bootcamp.database;

Output: MAX(age)

33

select MIN(age) from bootcamp.database;

Output: MIN(age)

22

select SUM(age) from bootcamp.database;

Output: SUM(age)

338

1. Using where clause for group by, order by.

select firstname,age from bootcamp.database group by firstname;

select firstname,count(\*) from bootcamp.database group by age;

select firstname, lastname, age, department from bootcamp.database order by firstname, lastname;

11.Difference between AND and OR Logical Operators.

The AND operator returns true if both expressions evaluate to true. If one of the two expressions is false, then the AND operator returns false even one of the expressions is NULL.

The SQL OR is a logical operator that combines two boolean expressions. The SQL OR operator returns either true or false depending on the results of expressions.

The OR operator is typically used in the WHERE clause of the SELECT, UPDATE, or DELETE statement to form a flexible condition.

12. Difference between and like :

The LIKE Operator in SQL is used to extract records where a particular pattern is present. In a WHERE clause, the LIKE operator is used to look for a certain pattern in a column

The BETWEEN operator in SQL is used to select values within a given range. For example, to extract only those records where the age of the person is between 20 and 25, we have to use the BETWEEN query in SQL.