Please create the database using project1.txt first

1. (in mysql workbench, the sql editor setting safe updates needs to be unchecked)

EXPLAIN SELECT \* FROM central\_Processing\_Unit WHERE cpuId = 1;

ALTER TABLE central\_Processing\_Unit DROP FOREIGN KEY central\_processing\_unit\_ibfk\_1;

ALTER TABLE ram\_speed\_junction DROP FOREIGN KEY ram\_speed\_junction\_ibfk\_1;

ALTER TABLE cpu\_socket\_type\_junction DROP FOREIGN KEY cpu\_socket\_type\_junction\_ibfk\_1;

ALTER TABLE central\_Processing\_Unit DROP PRIMARY KEY;

ALTER TABLE central\_Processing\_Unit

PARTITION BY RANGE(brandId)

(

PARTITION p0 VALUES LESS THAN (2),

PARTITION p1 VALUES LESS THAN (3),

PARTITION p2 VALUES LESS THAN (4),

PARTITION p3 VALUES LESS THAN (5),

PARTITION p4 VALUES LESS THAN (6),

PARTITION p5 VALUES LESS THAN (7),

PARTITION p6 VALUES LESS THAN (8),

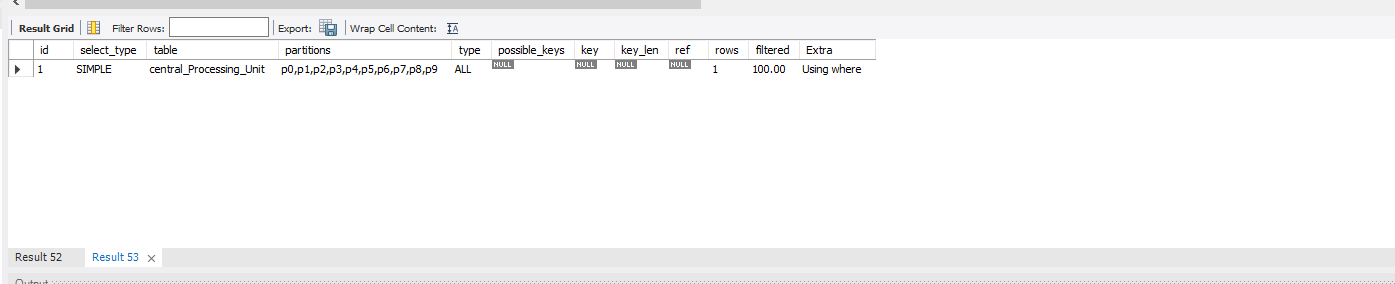
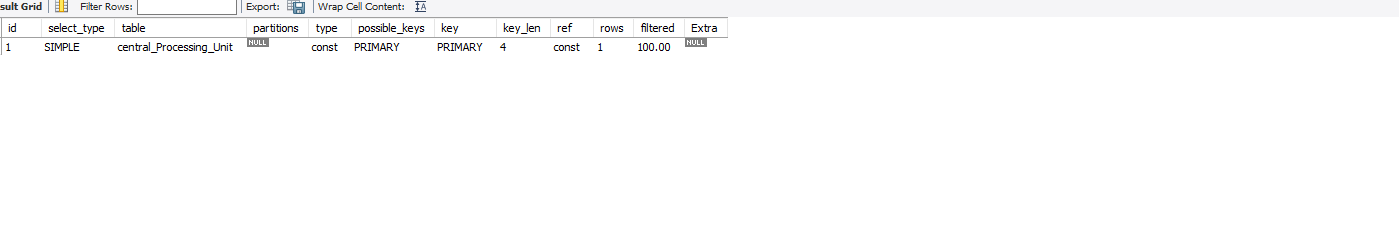
PARTITION p7 VALUES LESS THAN (9),

PARTITION p8 VALUES LESS THAN (10),

PARTITION p9 VALUES LESS THAN MAXVALUE

);

EXPLAIN SELECT \* FROM central\_Processing\_Unit WHERE cpuId = 1;





DELIMITER $$

CREATE PROCEDURE GetCasesByBrand(IN brandId INT)

BEGIN

SELECT \*

FROM computerCase

WHERE computerCase.brandId = brandId;

END$$

DELIMITER ;

CALL GetCasesByBrand(3);



CREATE VIEW brand\_view AS SELECT \* FROM brand;

CREATE VIEW color\_view AS SELECT \* FROM color;

CREATE VIEW formfactor\_view AS SELECT \* FROM formfactor;

CREATE VIEW computerCase\_view AS SELECT \* FROM computerCase;

CREATE VIEW powersupply\_view AS SELECT \* FROM powersupply;

CREATE VIEW cpu\_socket\_type\_view AS SELECT \* FROM cpu\_socket\_type;

CREATE VIEW central\_Processing\_Unit\_view AS SELECT \* FROM central\_Processing\_Unit;

CREATE VIEW motherboard\_view AS SELECT \* FROM motherboard;

CREATE VIEW gaphics\_Processing\_Unit\_view AS SELECT \* FROM gaphics\_Processing\_Unit;

CREATE VIEW ram\_Speed\_Type\_view AS SELECT \* FROM ram\_Speed\_Type;

CREATE VIEW ram\_view AS SELECT \* FROM ram;

CREATE VIEW hard\_disk\_view AS SELECT \* FROM hard\_disk;

CREATE VIEW caseFormFactor\_junction\_view AS SELECT \* FROM caseFormFactor\_junction;

CREATE VIEW ram\_Speed\_junction\_view AS SELECT \* FROM ram\_Speed\_junction;

CREATE VIEW ram\_motherboard\_junction\_view AS SELECT \* FROM ram\_motherboard\_junction;

CREATE VIEW cpu\_socket\_type\_junction\_view AS SELECT \* FROM cpu\_socket\_type\_junction;

CREATE ROLE ViewAll;

GRANT SELECT ON brand TO ViewAll;

GRANT SELECT ON color TO ViewAll;

GRANT SELECT ON formfactor TO ViewAll;

GRANT SELECT ON computerCase TO ViewAll;

GRANT SELECT ON powersupply TO ViewAll;

GRANT SELECT ON cpu\_socket\_type TO ViewAll;

GRANT SELECT ON central\_Processing\_Unit TO ViewAll;

GRANT SELECT ON motherboard TO ViewAll;

GRANT SELECT ON gaphics\_Processing\_Unit TO ViewAll;

GRANT SELECT ON ram\_Speed\_Type TO ViewAll;

GRANT SELECT ON ram TO ViewAll;

GRANT SELECT ON hard\_disk TO ViewAll;

GRANT SELECT ON caseFormFactor\_junction TO ViewAll;

GRANT SELECT ON ram\_Speed\_junction TO ViewAll;

GRANT SELECT ON ram\_motherboard\_junction TO ViewAll;

GRANT SELECT ON cpu\_socket\_type\_junction TO ViewAll;

CREATE USER 'viewuser'@'localhost' IDENTIFIED BY 'password';

GRANT ViewAll TO 'viewuser'@'localhost';



CREATE ROLE AppAdmin;

GRANT SELECT, INSERT, UPDATE, DELETE, REFERENCES ON \*.\* TO AppAdmin;

CREATE USER 'appuser'@'localhost' IDENTIFIED BY 'password';

GRANT AppAdmin TO 'appuser'@'localhost';



ALTER TABLE computerCase

ADD COLUMN front\_panel\_usb\_c TINYINT UNSIGNED;

ALTER TABLE computerCase

RENAME TO pc\_case;



This was already done when creating the table in readme

1. (pc\_case table is what is changed from computercase per 5)

CREATE TABLE pc\_case\_history (

id INT AUTO\_INCREMENT PRIMARY KEY,

computerCaseId INT,

brandId INT,

colorId INT,

num\_120fans INT,

caseName VARCHAR(100),

changed\_by VARCHAR(100),

changed\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

change\_type VARCHAR(10)

);

DELIMITER $$

CREATE TRIGGER pc\_case\_update\_trigger

BEFORE UPDATE ON pc\_case

FOR EACH ROW

BEGIN

INSERT INTO pc\_case\_history

(computerCaseId, brandId, colorId, num\_120fans, caseName, changed\_by, change\_type)

VALUES

(OLD.computerCaseId, OLD.brandId, OLD.colorId, OLD.num\_120fans, OLD.caseName, CURRENT\_USER(), 'OLD');

INSERT INTO pc\_case\_history

(computerCaseId, brandId, colorId, num\_120fans, caseName, changed\_by, change\_type)

VALUES

(NEW.computerCaseId, NEW.brandId, NEW.colorId, NEW.num\_120fans, NEW.caseName, CURRENT\_USER(), 'NEW');

END$$

DELIMITER ;



ALTER TABLE pc\_case

ADD CONSTRAINT case\_fans\_nonnegative

CHECK (num\_120fans >= 0);

ALTER TABLE powersupply

ADD CONSTRAINT psu\_wattage\_range

CHECK (wattageOut BETWEEN 500 AND 1500);



CREATE SCHEMA copy\_schema;

CREATE TABLE copy\_schema.pc\_case AS

SELECT \* FROM pc\_case;

INSERT INTO copy\_schema.pc\_case

SELECT \* FROM pc\_case;

CREATE USER 'new\_user'@'localhost' IDENTIFIED BY 'password';

GRANT SELECT, REFERENCES, UPDATE, DELETE

ON copy\_schema.pc\_case

TO 'new\_user'@'localhost';

1. (Please change <Your username> and < Your password> in these codes)

The code below would find motherboards with DDR4-3200 capability and cpu that matches the motherboard and the capability

import mysql.connector

myquery = "SELECT r.ramName, mb.motherboardName, cpu.cpuName FROM ram r JOIN ram\_motherboard\_junction rmj ON r.TypeId = rmj.TypeId JOIN motherboard mb ON mb.motherboardId = rmj.motherboardId JOIN cpu\_socket\_type\_junction cstj ON mb.motherboardId = cstj.motherboardId JOIN central\_Processing\_Unit cpu ON cpu.cpuId = cstj.TypeId JOIN ram\_speed\_junction rsj ON r.TypeId = rsj.TypeId WHERE r.TypeId = (SELECT TypeId FROM ram\_Speed\_Type WHERE TypeName = 'DDR4-3200') AND cpu.cpuId = rsj.cpuId;"

try:

cnx = mysql.connector.connect(user='homaxx', password='Blueocean1005!',

host='127.0.0.1',database='project1')

# Prepare a cursor object using cursor() method

cursor = cnx.cursor ()

# Execute the SQL query using execute() method.

cursor.execute (myquery)

# Fetch all rows

myresult = cursor.fetchall()

for x in myresult:

print(x)

except mysql.connector.Error as e:

print (e)

finally:

# Close the cursor object

cursor.close ()

# Close the connection

cnx.close ()