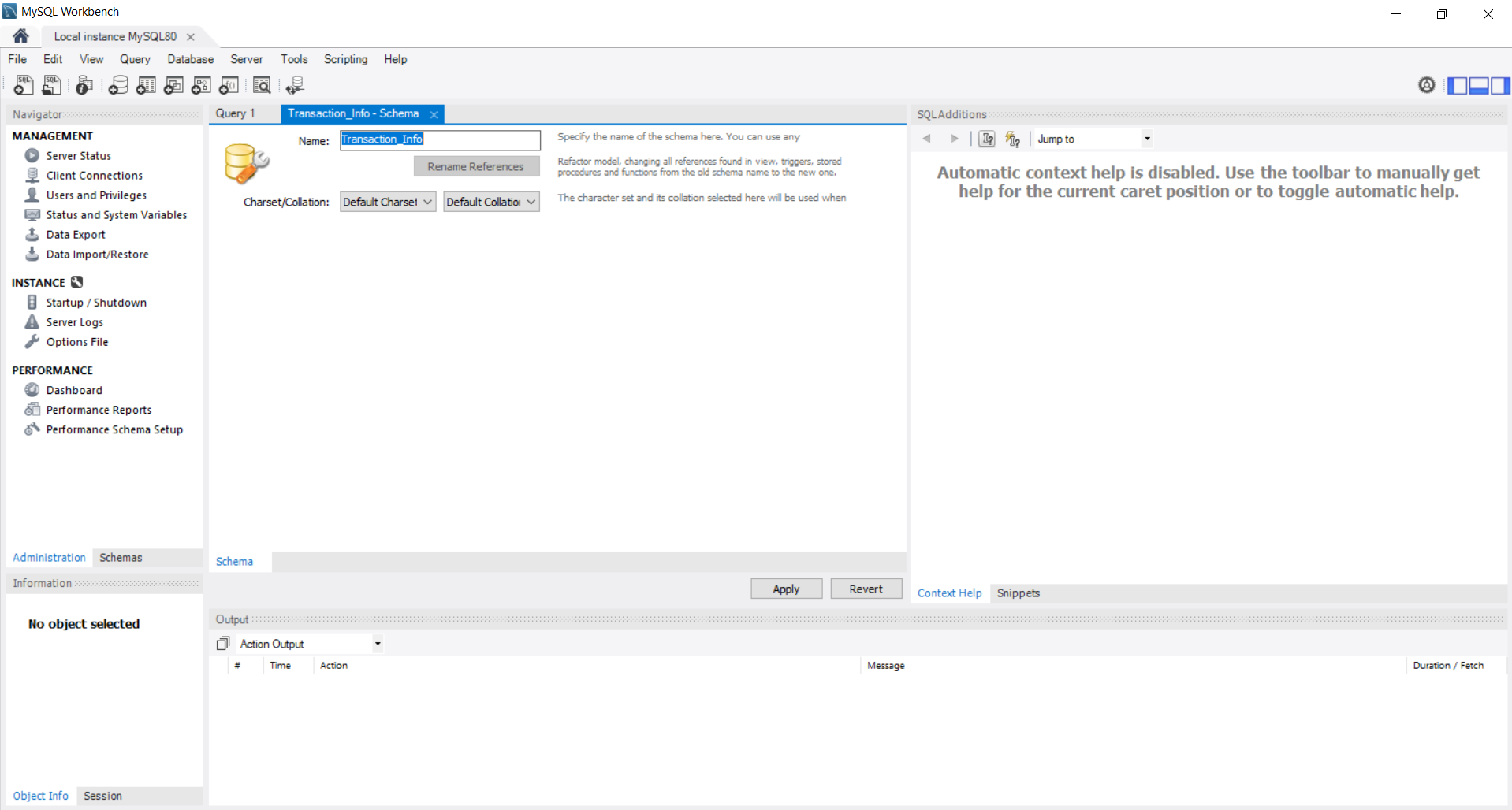
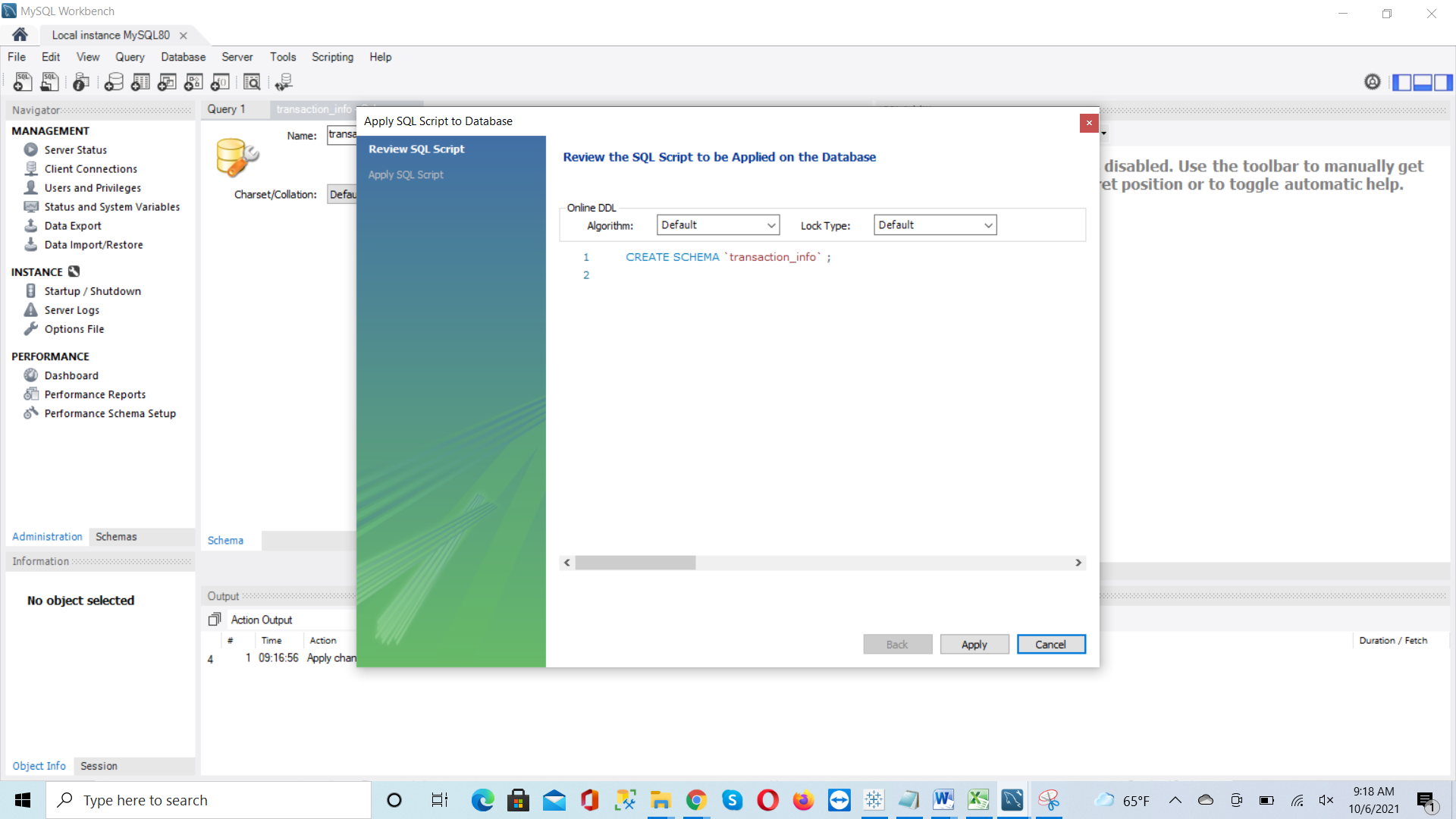
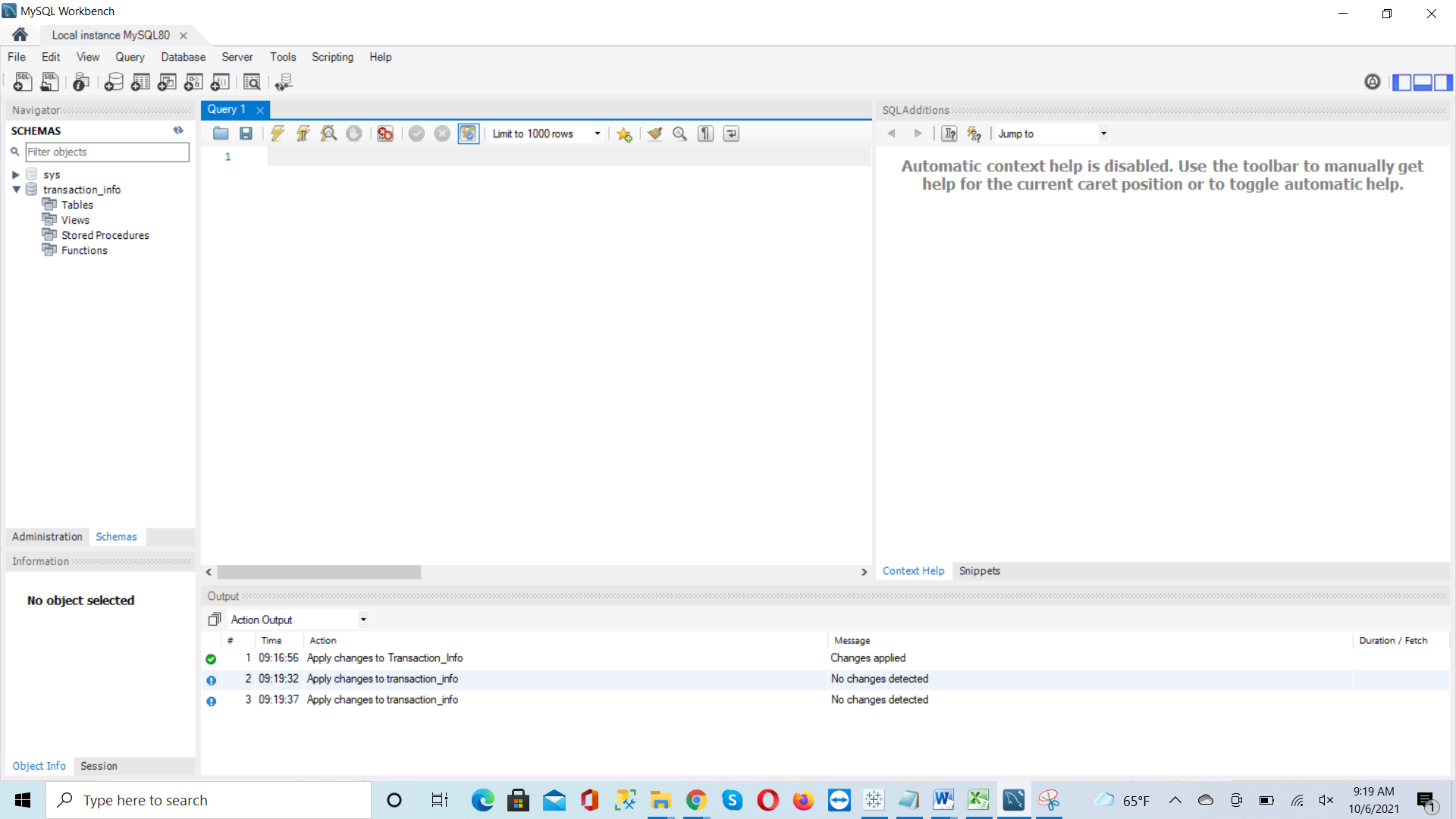
**Creating the DB in MySQL**

**(transaction\_info)**







**Create tables:**

**(account\_info)**

CREATE TABLE `transaction\_info`.`account\_info` (

`first\_name` VARCHAR(100) NOT NULL,

`last\_name` VARCHAR(100) NOT NULL,

`street\_address` VARCHAR(200) NOT NULL,

`unit` VARCHAR(100) NULL,

`city` VARCHAR(100) NOT NULL,

`state` VARCHAR(100) NOT NULL,

`zipcode` VARCHAR(100) NOT NULL,

`dob` DATE NOT NULL,

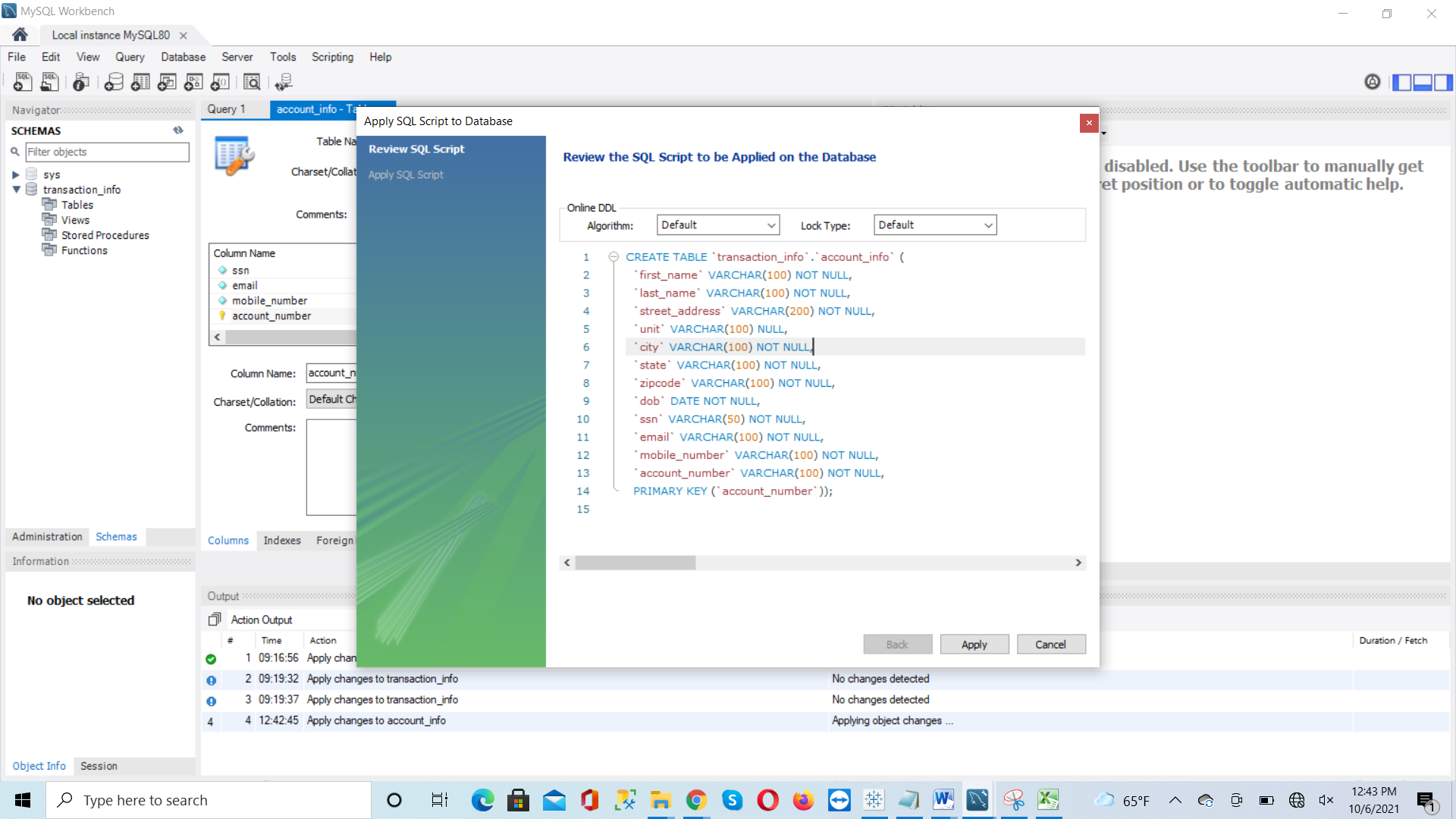
`ssn` VARCHAR(50) NOT NULL,

`email` VARCHAR(100) NOT NULL,

`mobile\_number` VARCHAR(100) NOT NULL,

`account\_number` VARCHAR(100) NOT NULL,

PRIMARY KEY (`account\_number`));



**To change the dob column Date type from date to VARCHAR(50):**

ALTER TABLE `transaction\_info`.`account\_info`

CHANGE COLUMN `dob` `dob` VARCHAR(50) NOT NULL ;

**Second Table:**

**(transaction)**

CREATE TABLE `transaction\_info`.`transaction` (

`account\_number` VARCHAR(100) NOT NULL,

`merchant` VARCHAR(100) NOT NULL,

`merchant\_category\_code` VARCHAR(100) NOT NULL,

`merchant\_number` VARCHAR(100) NOT NULL,

`post\_date` VARCHAR(100) NOT NULL,

`transaction\_amount` FLOAT NOT NULL,

`transaction\_date` VARCHAR(100) NOT NULL,

`transaction\_number` INT NOT NULL,

`transaction\_state` VARCHAR(100) NOT NULL,

PRIMARY KEY (`account\_number`, `transaction\_number`),

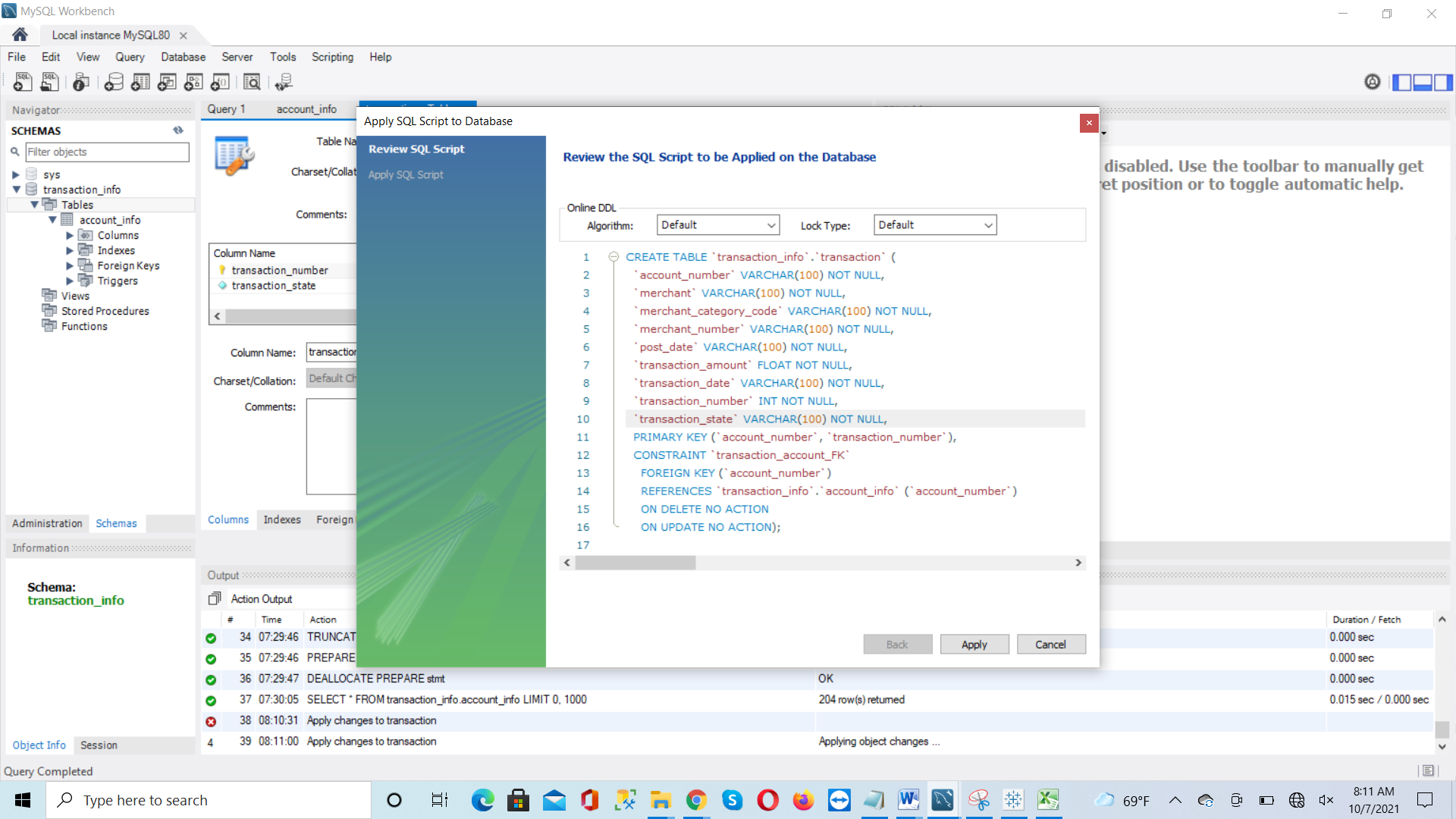
CONSTRAINT `transaction\_account\_FK`

FOREIGN KEY (`account\_number`)

REFERENCES `transaction\_info`.`account\_info` (`account\_number`)

ON DELETE NO ACTION

ON UPDATE NO ACTION);



**Rule#2:**

use transaction\_info;

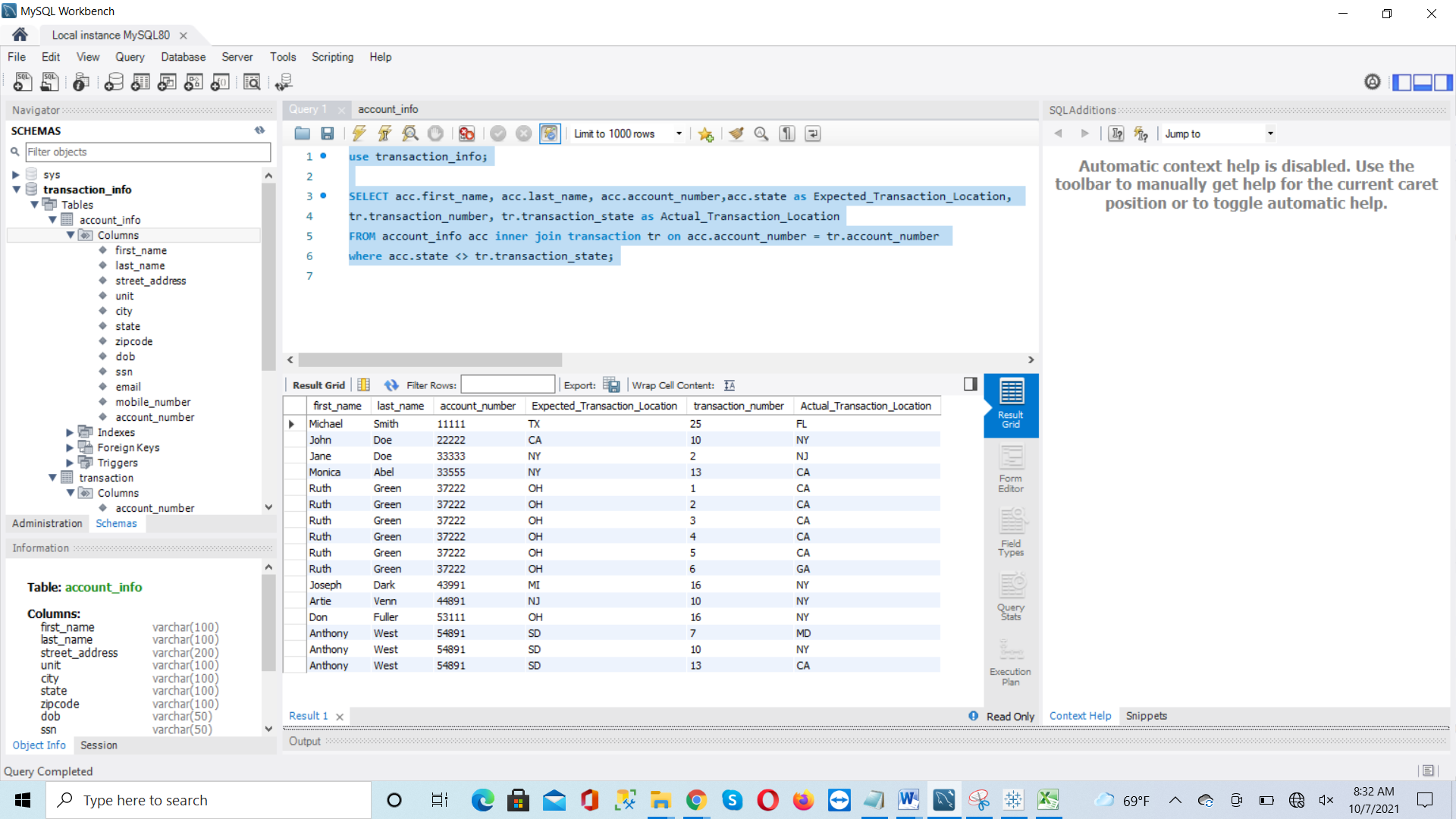
SELECT acc.first\_name, acc.last\_name, acc.account\_number,acc.state as

Expected\_Transaction\_Location, tr.transaction\_number, tr.transaction\_state as

Actual\_Transaction\_Location

FROM account\_info acc inner join transaction tr on acc.account\_number = tr.account\_number

WHERE acc.state <> tr.transaction\_state;



**Creates View for Z-Score:**

**(View: process\_zscore)**

CREATE VIEW transaction\_info.process\_zscore as

SELECT tr.account\_number, tr.merchant, tr.transaction\_number, tr.transaction\_amount,

AVG(tr.transaction\_amount) as avg\_transaction, STD(tr.transaction\_amount) as

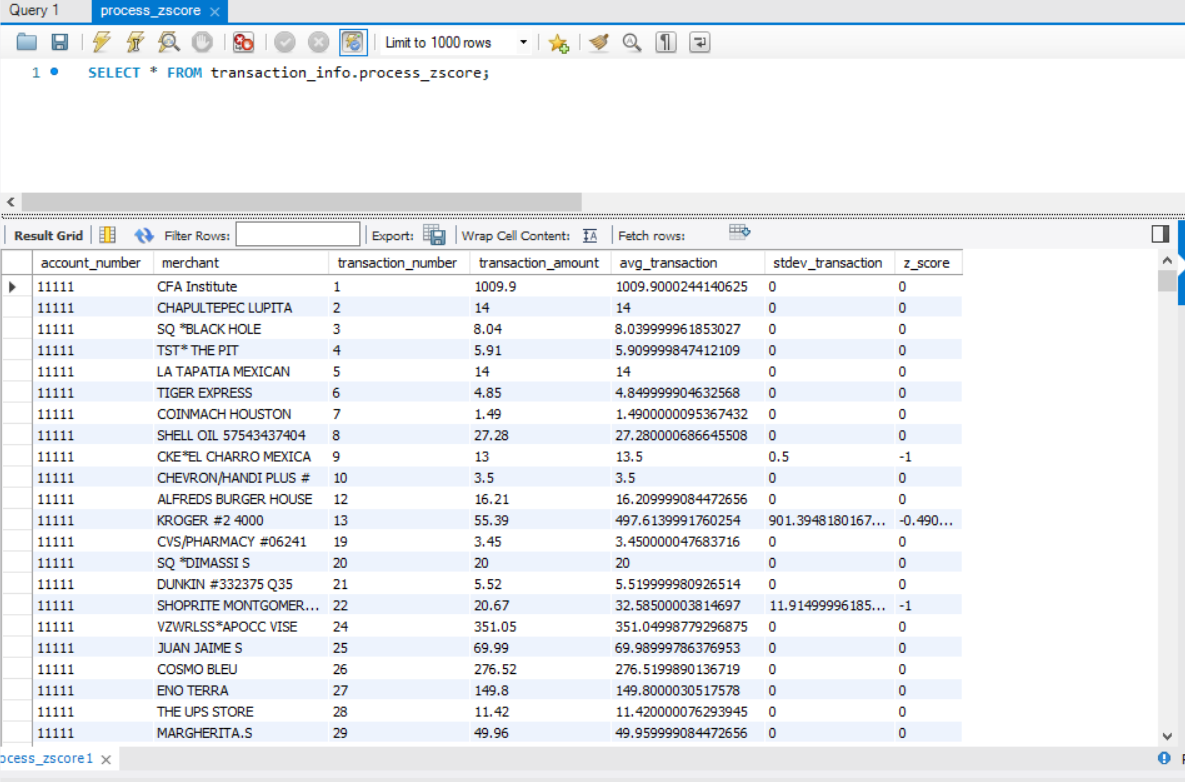
stdev\_transaction, if (STD(tr.transaction\_amount)<>0, (transaction\_amount –

AVG(tr.transaction\_amount))/ STD(tr.transaction\_amount), 0) as z\_score

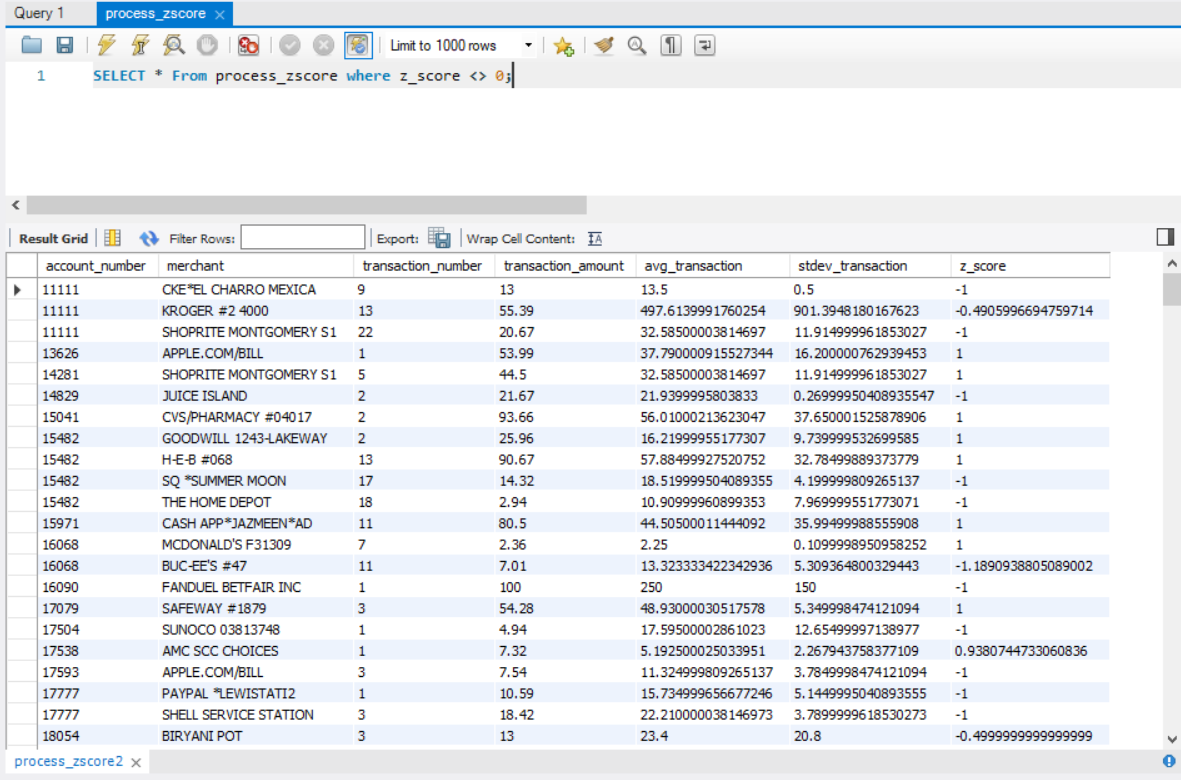
FROM transaction tr GROUP BY tr.merchant, tr.account\_number;

**Screenshot for the view:**

SELECT \* FROM transaction\_info.process\_zscore;



SELECT \* From process\_zscore where z\_score <> 0;



**Create z-score with account information:**

**(View: process\_zscore\_account)**

use transaction\_info;

CREATE VIEW transaction\_info.process\_zscore\_account as

SELECT CONCAT(CONCAT(acc.first\_name, " "), acc.last\_name) as customer\_name,

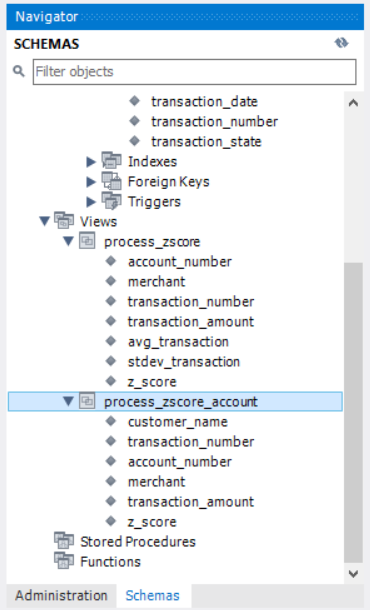
zs.transaction\_number, acc.account\_number, zs.merchant, zs.transaction\_amount,

zs.z\_score

FROM account\_info acc inner join process\_zscore zs on

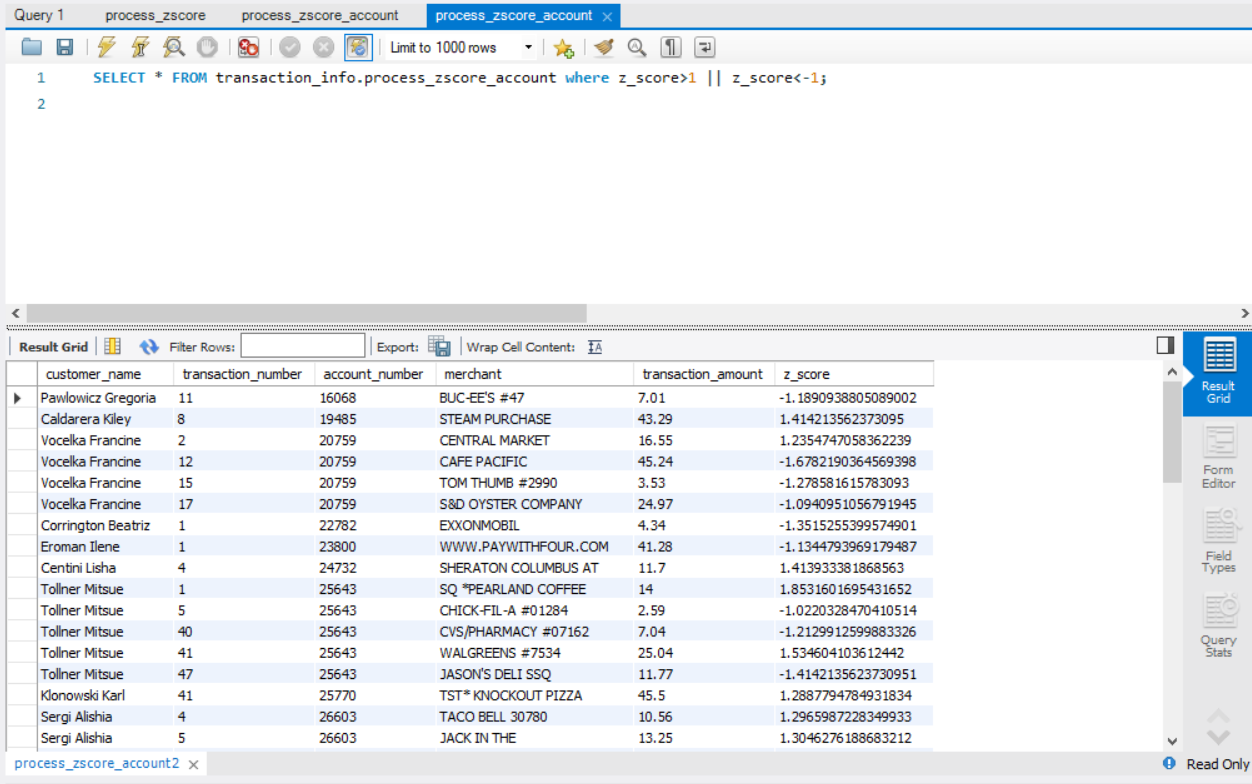
acc.account\_number = zs.account\_number

**Screenshot for the two views created:**

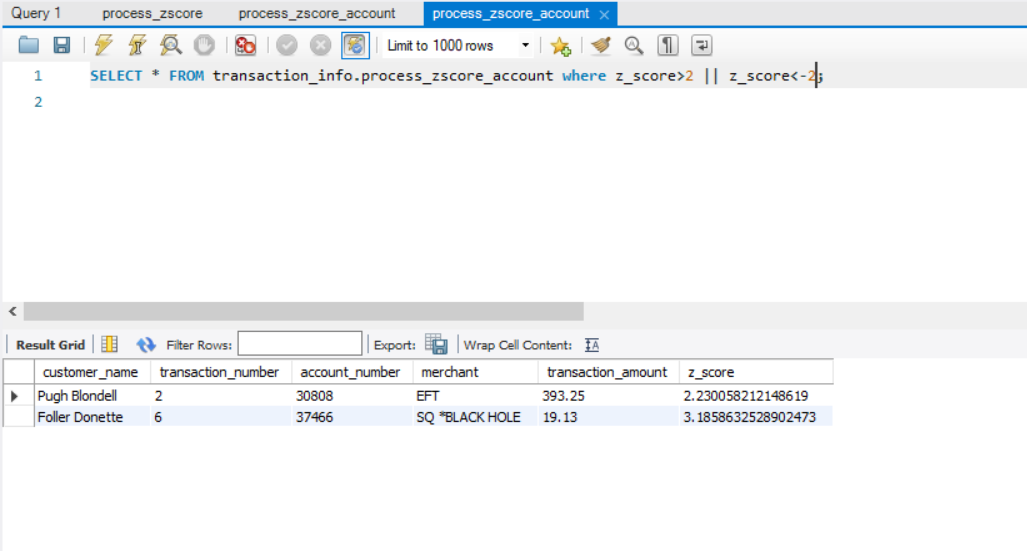


**Pulling out data from the view:**

SELECT \* FROM transaction\_info.process\_zscore\_account where z\_score>1 || z\_score<-1;



SELECT \* FROM transaction\_info.process\_zscore\_account where z\_score>2 || z\_score<-2;



**RULE #1:**

SELECT customer\_name,transaction\_number, account\_number, merchant,

CONCAT("$",transaction\_amount) as transaction\_amount, z\_score

FROM transaction\_info.process\_zscore\_account where z\_score>2 || z\_score<-2;

**Screenshot:**

