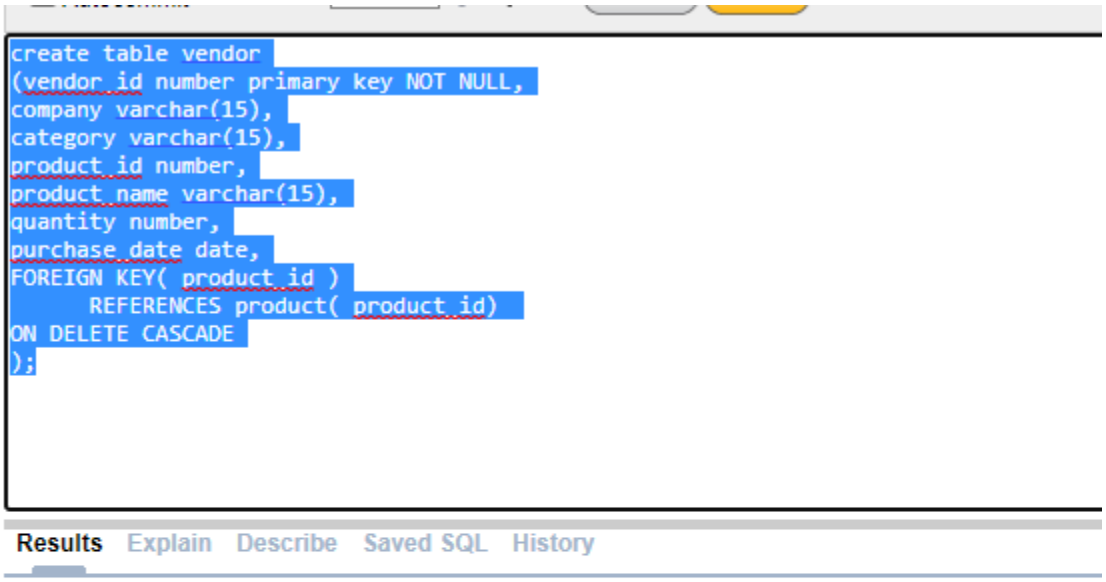
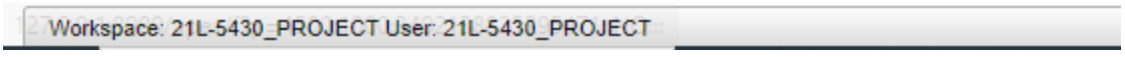
Vendors:

```
create table vendor
(vendor_id number primary key NOT NULL,
company varchar(15),
category varchar(15),
product_id number,
product_name varchar(15),
quantity number,
purchase_date date,
FOREIGN KEY( product_id )
     REFERENCES product( product_id)
ON DELETE CASCADE
);
```





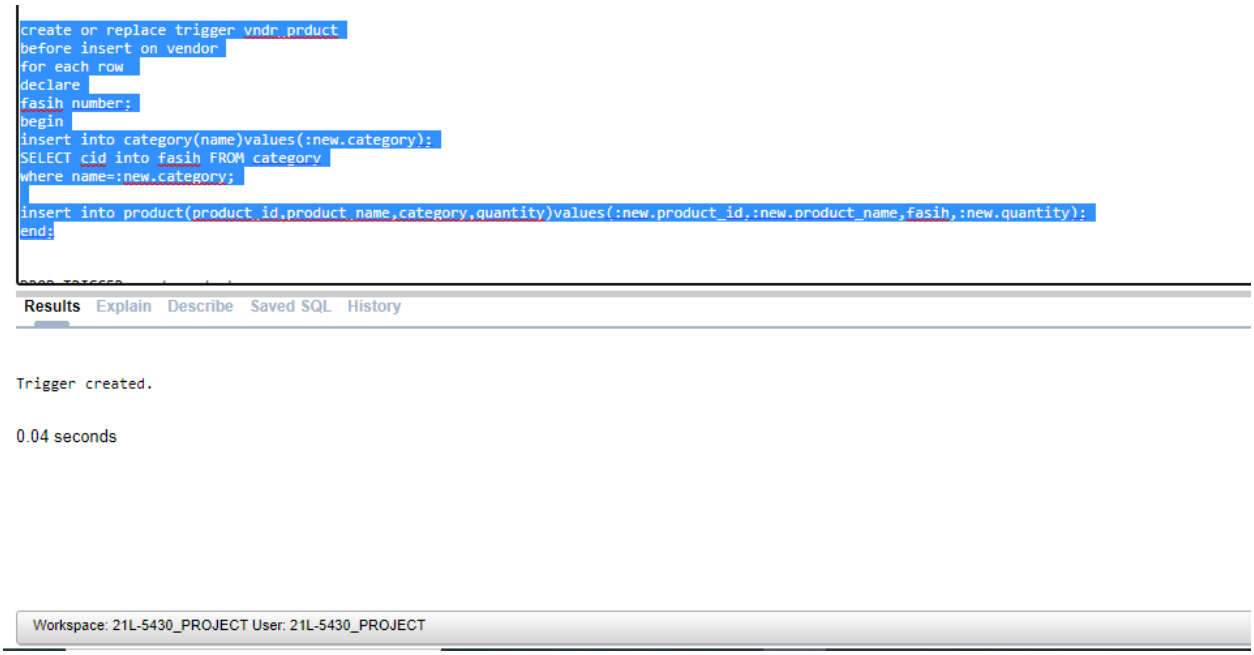
Trigger for vendor insert

```
create or replace trigger vndr_prduct
before insert on vendor
for each row
declare
fasih number;
checking number;
checking1 number;
begin
select count(name) into checking
from category
where name = :new.category;
if checking = 0
then
insert into category(name)values(:new.category);
end if;
select count(product_id) into checking1
from product
where  product_id = :new.product_id;
if checking1 = 0
then
SELECT cid into fasih FROM category
where name=:new.category;
insert into
product(product_id,product_name,category,quantity)values(:new.product_id,:new.produ
ct_name,fasih,:new.quantity);
else
update product
set quantity=quantity + :new.quantity
where product_id = :new.product_id;
end if;
end;
```

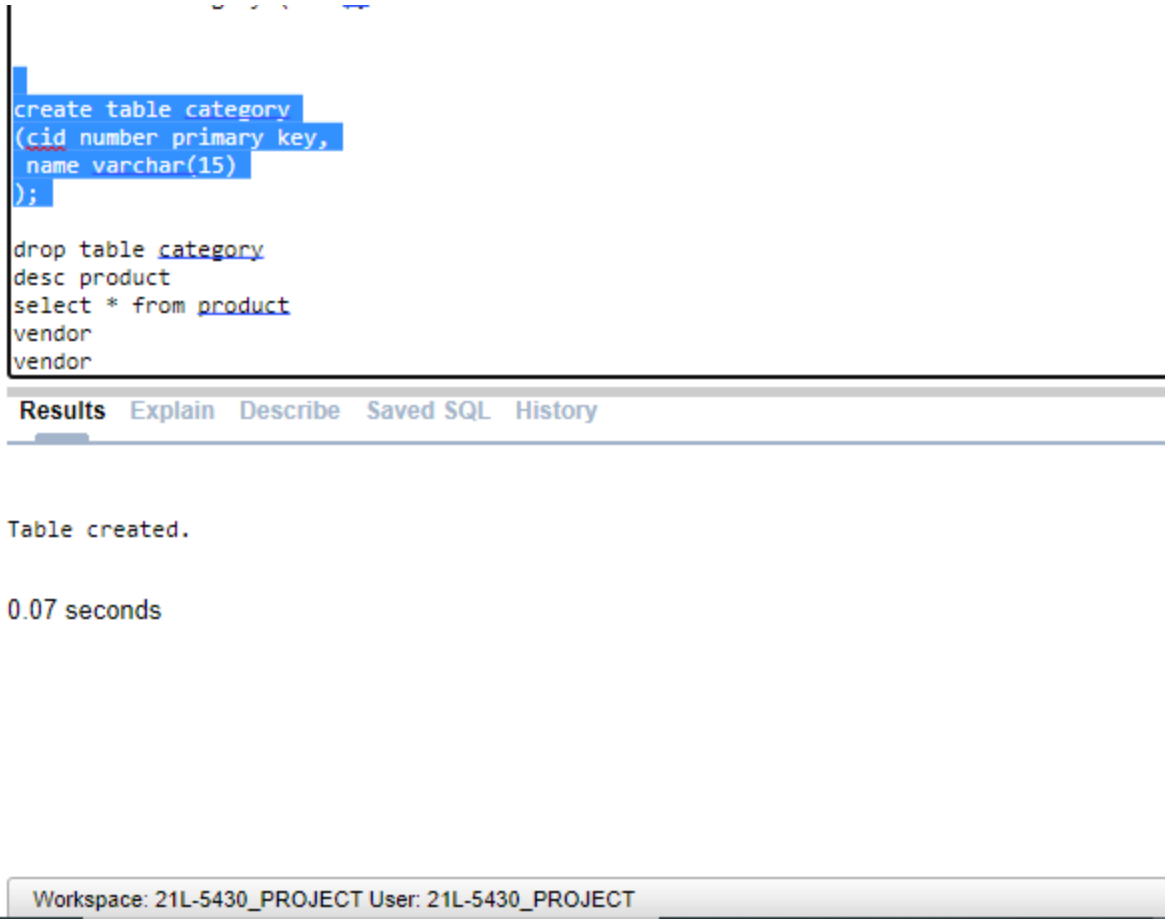
Trigger for vendor update:

create or replace trigger vndr_prduct
before update on vendor
for each row
declare
fasih number;
begin
insert into category(name)values(:new.category);
SELECT cid into fasih FROM category
where name=:new.category;

insert into
product(product_id,product_name,category,quantity)values(:new.product_id,:new
.product_name,fasih,:new.quantity);
end;



Category table:



Category:

```
CREATE TABLE category (
  cid NUMBER PRIMARY KEY,
  name VARCHAR2(15) NOT NULL
);
```

Sequence for cid

```
CREATE SEQUENCE cat_seq START WITH 1;
select SEQUENCE
desc category
CREATE OR REPLACE TRIGGER category_trg
BEFORE INSERT ON category
FOR EACH ROW
BEGIN
  :NEW.cid := cat_seq.NEXTVAL;
end;
/
```

```
name varchar(15)
);
CREATE SEQUENCE cid_seq START WITH 1 INCREMENT BY 1;

CREATE TABLE category
(
    cid NUMBER DEFAULT cid_seq.nextval PRIMARY KEY,
    name VARCHAR2(15) NOT NULL
);
```

Results Explain Describe Saved SQL History

```
Sequence created.
```

0.01 seconds

Workspace: 21L-5430_PROJECT User: 21L-5430_PROJECT

```
desc category
CREATE OR REPLACE TRIGGER category_trg
BEFORE INSERT ON category
FOR EACH ROW
BEGIN
  :NEW.cid_:= cat_seq.NEXTVAL;
end;
/



drop table category
desc product
select * from product
vendor
vendor
create or replace trigger vndr_update
```

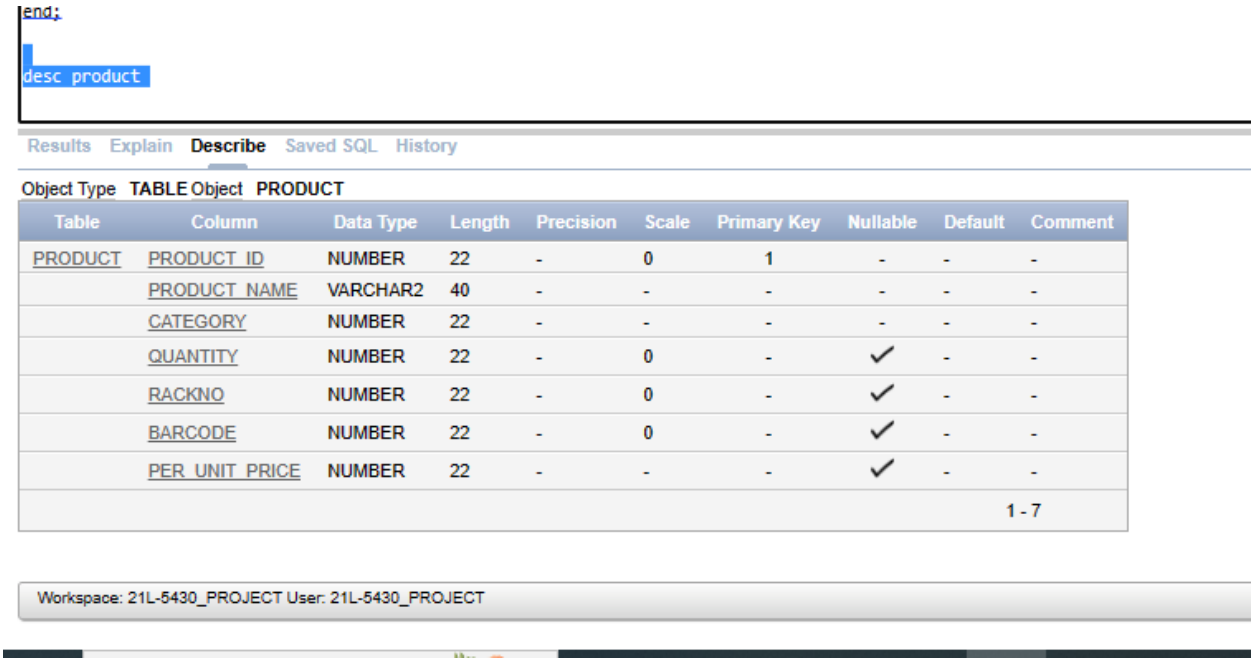
Results Explain Describe Saved SQL History

```
Trigger created.
```

0.13 seconds

Workspace: 21L-5430_PROJECT User: 21L-5430_PROJECT

Product table:



```
end;

desc product
```

Results Explain **Describe** Saved SQL History

Object Type **TABLE** Object **PRODUCT**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PRODUCT	PRODUCT_ID	NUMBER	22	-	0	1	-	-	-
	PRODUCT_NAME	VARCHAR2	40	-	-	-	-	-	-
	CATEGORY	NUMBER	22	-	-	-	-	-	-
	QUANTITY	NUMBER	22	-	0	-	✓	-	-
	RACKNO	NUMBER	22	-	0	-	✓	-	-
	BARCODE	NUMBER	22	-	0	-	✓	-	-
	PER_UNIT_PRICE	NUMBER	22	-	-	-	✓	-	-

1 - 7

Workspace: 21L-5430_PROJECT User: 21L-5430_PROJECT

```
ALTER TABLE product
ADD CONSTRAINT fk_prod_cat FOREIGN KEY (category)
REFERENCES category (cid)  on delete cascade;
```

```
ALTER TABLE product
ADD CONSTRAINT fk_prod_cat FOREIGN KEY (category)
REFERENCES category (cid)  on delete cascade;
```

Results Explain Describe Saved SQL History

```
Table altered.
```

0.04 seconds

Workspace: 21L-5430_PROJECT User: 21L-5430_PROJECT

Sales table:

```
CREATE TABLE sales (
   customer_id number,
   product_id number,
   quantity number,
   per_unit_price number,
   total number,
   CONSTRAINT fk_sal_cust FOREIGN KEY (customer_id)
      REFERENCES customer(customer_id) ON DELETE CASCADE,
   CONSTRAINT fk_sal_prod FOREIGN KEY (product_id)
      REFERENCES product(product_id)
```

);

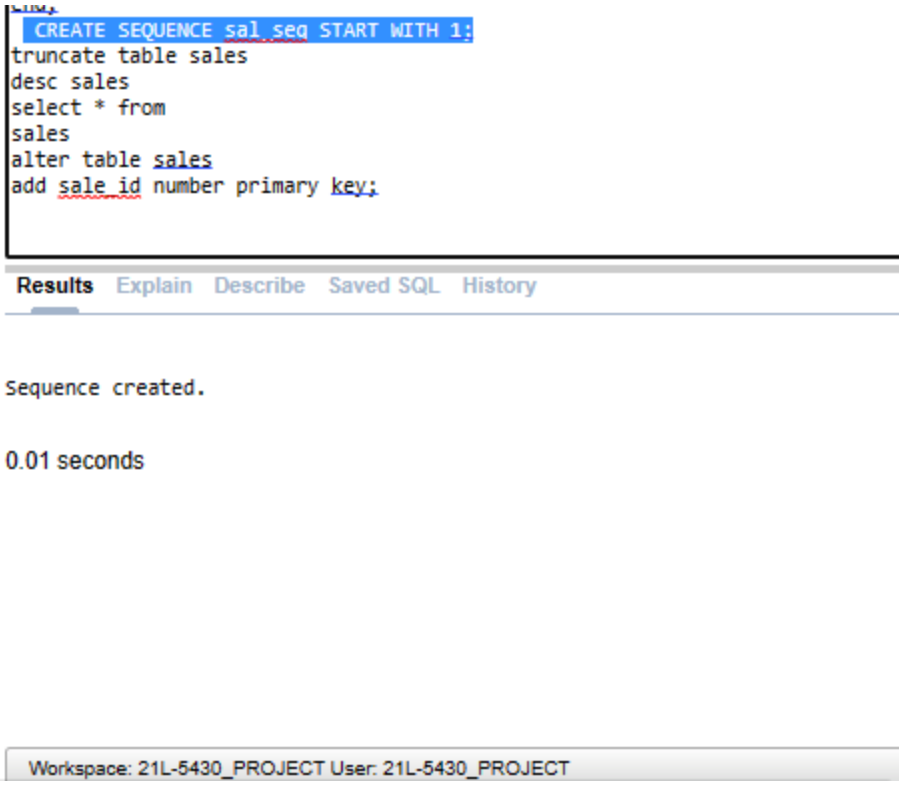
```
CREATE TABLE sales (
    customer_id number,
    product_id number,
    quantity number,
    per_unit_price number,
    total number,
    CONSTRAINT fk_sal_cust FOREIGN KEY (customer_id)
        REFERENCES customer(customer_id) ON DELETE CASCADE,
    CONSTRAINT fk_sal_prod FOREIGN KEY (product_id)
        REFERENCES product(product_id)
);

)
```

Results Explain Describe Saved SQL History

```
Table created.
```

0.28 seconds

Workspace: 21L-5430_PROJECT User: 21L-5430_PROJECT



Created sequence of for sid

```
CREATE OR REPLACE TRIGGER sale_trg
BEFORE INSERT ON sales
FOR EACH ROW
BEGIN
  :NEW.sale_id := sal_seq.NEXTVAL;
end;
```

```
add sale_id number primary key;
CREATE OR REPLACE TRIGGER sale_trg
BEFORE INSERT ON sales
FOR EACH ROW
BEGIN
  :NEW.sale_id := sal_seq.NEXTVAL;
end;
```

Results Explain Describe Saved SQL History

```
Trigger created.
```

0.10 seconds

Workspace: 21L-5430_PROJECT User: 21L-5430_PROJECT

Trigger on sales table:

```
create or replace trigger sal_info
before insert on sales
for each row
declare
checking number;
begin
select count(customer_id) into checking
from customer
where customer_id = :new.customer_id;
if checking = 0
then
insert into customer (customer_id) values(:new.customer_id);
end if;
update product
set quantity=quantity - :new.quantity
where (product_id= :new.product_id) and (quantity >= :new.quantity);
end;
```

```
drop trigger sal_info
create or replace trigger sal_info
before insert on sales
for each row
declare
checking number;
begin
select count(customer_id) into checking
from customer
where customer_id = :new.customer_id;
if checking = 0
then
insert into customer (customer_id) values(:new.customer_id);
end if;
update product
set quantity=quantity - :new.quantity
where (product_id= :new.product_id) and (quantity >= :new.quantity);
end;
```

Results Explain Describe Saved SQL History

```
Trigger created.
```

0.05 seconds

Workspace: 21L-5430_PROJECT User: 21L-5430_PROJECT

Account table:

```
create table account (
account_id number primary key,
sid number,
customer_id number,
revenue number,
expense number,
profit number,
constraint fk__sal_account foreign key (sid)
references sales(sale_id)
);
```

```
  CREATE SEQUENCE acc_seq START WITH 1;
truncate table sales
desc sales
select * from
sales
alter table sales
add sale_id number primary key;
```

Results Explain Describe Saved SQL History

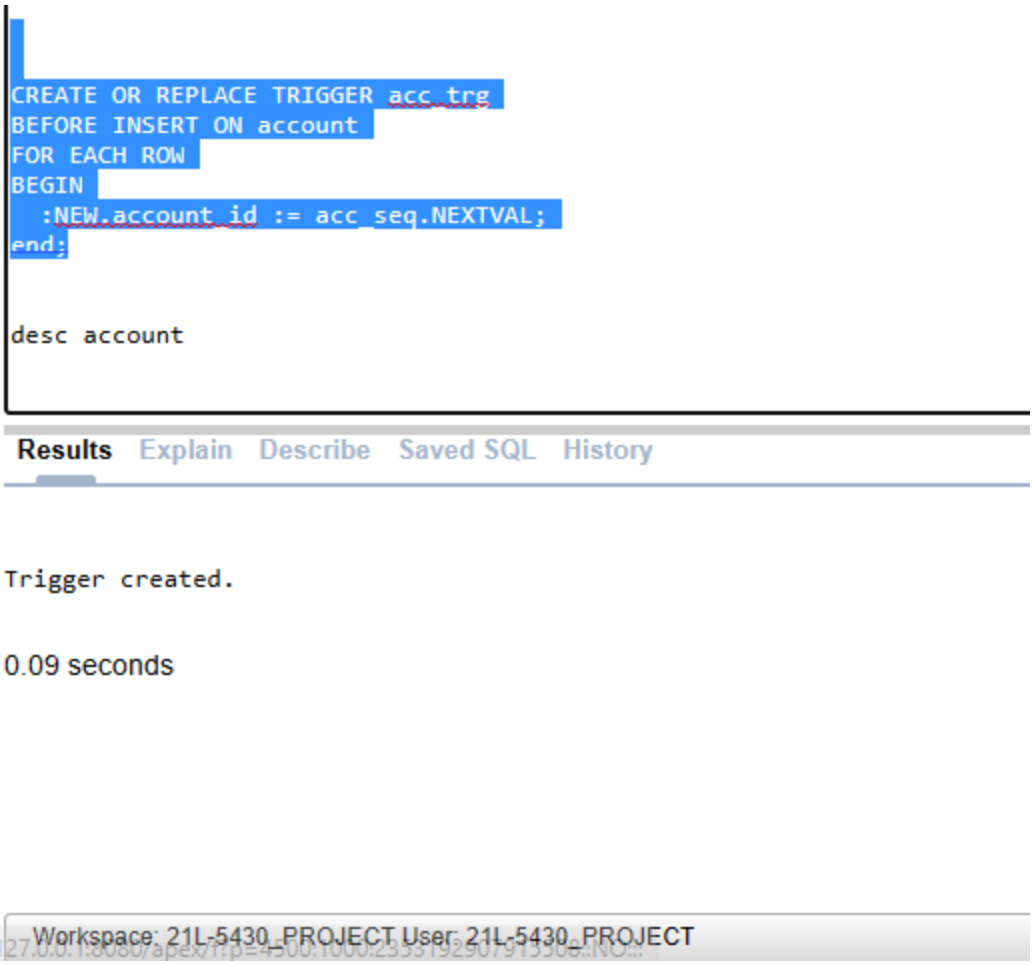
```
Sequence created.
```

0.02 seconds

127.0.0.1:8080/apex/f?p=4500:1000:2353192907915508::NO::: :CT

Sequence for account_id

```
CREATE OR REPLACE TRIGGER acc_trg
BEFORE INSERT ON account
FOR EACH ROW
BEGIN
  :NEW.account_id := acc_seq.NEXTVAL;
end;
```



Trigger on sale insert

```
CREATE OR REPLACE TRIGGER fk_acc_sal
after INSERT ON sales
FOR EACH ROW
DECLARE
  exp float:=:new.per_unit_price;
  prof float;
BEGIN
--DBMS_output.put_line(exp);
exp := exp*0.1;
exp:=exp* :new.quantity;
  exp := :new.total - exp;
  prof := :new.total - exp;
  INSERT into account(sid, customer_id, revenue, expense, profit)
  VALUES (:new.sale_id, :new.customer_id, :new.total, exp, prof);
```

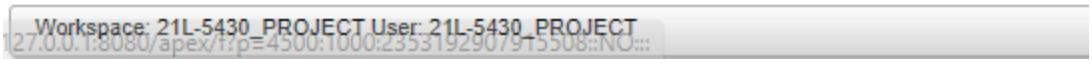
END;

```
);
CREATE OR REPLACE TRIGGER fk_acc_sal
after INSERT ON sales
FOR EACH ROW
DECLARE
  exp float:=:new.per_unit_price;
  prof float;
BEGIN
--DBMS_output.put_line(exp);
exp := exp*0.1;
exp:=exp* :new.quantity;
  exp := :new.total - exp;
  prof := :new.total - exp;
  INSERT into account(sid, customer_id, revenue, expense, profit)
  VALUES (:new.sale_id, :new.customer_id, :new.total, exp, prof);
END;
```

Results Explain Describe Saved SQL History

```
Trigger created.
```

0.02 seconds



DML queries:

INSERT INTO sales (customer_id,product_id,quantity,per_unit_price,total) VALUES('" + cid.Text + "','" + pid.Text + "','" + quantity.Text + "','" + perprice.Text + "','" + total.Text + "')";

SELECT per_unit_price FROM product WHERE product_id = :productId

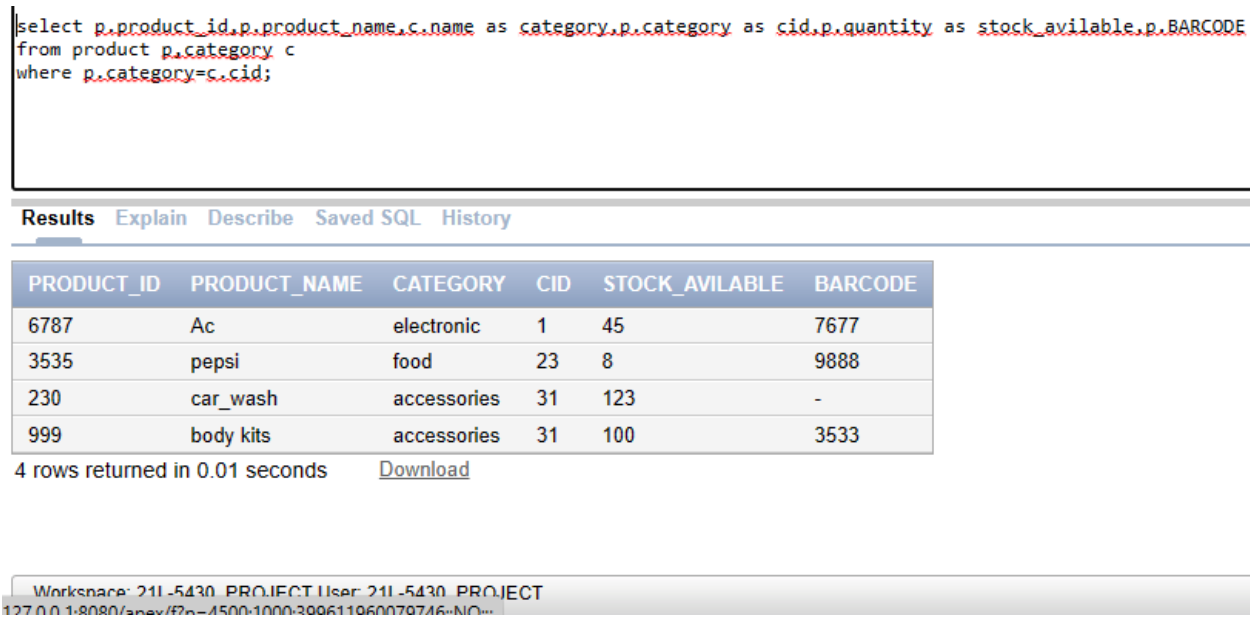
SELECT * FROM account

select * from product where quantity<10

INSERT INTO product VALUES('" + id.Text + "','" + name.Text + "','" + comboBox1.SelectedValue + "','" + quantity.Text + "','" + rack.Text + "','" + barcode.Text + "','" + price.Text + "')";

Use of joins to display inventory lvl:

select p.product_id,p.product_name,c.name as category,p.category as cid,p.quantity as stock_avilable,p.BARCODE
from product p,category c
where p.category=c.cid;



```
select p.product_id,p.product_name,c.name as category,p.category as cid,p.quantity as stock_avilable,p.BARCODE
from product p,category c
where p.category=c.cid;
```

Results Explain Describe Saved SQL History

PRODUCT_ID	PRODUCT_NAME	CATEGORY	CID	STOCK_AVILABLE	BARCODE
6787	Ac	electronic	1	45	7677
3535	pepsi	food	23	8	9888
230	car_wash	accessories	31	123	-
999	body kits	accessories	31	100	3533

4 rows returned in 0.01 seconds Download

Workspace: 21L-5430_PROJECT User: 21L-5430_PROJECT

For view Reports:

select p.product_id,p.product_name,c.name as category,p.category as cid,p.quantity as stock_avilable,p.BARCODE
from product p,category c
where p.category=c.cid;