Proiect Baze de Date

Stanciu Andreea-Maria, 244

Baza de date pe care am conceput-o are ca scop gestionarea comenzilor si a livrarii produselor comandate de catre un client. In aceasta baza de date, nu se pune accent pe modalitatea de plata sau pe modalitatea de efectuare a comenzii, ci strict pe procesul comanda - selectare din stoc - livrare produs(e).

In acest scop, ne vom folosi in principal de o entitate “client”, care efectueaza comanda, o entitate “comanda”, care gestioneaza comanda, o entitate “angajat”, care se ocupa de livrarea comenzii, o entitate “magazin”, care gestioneaza angajatii si produsele, si o entitate “produs”, care defineste produsul comandat.

Totusi, ne vom folosi si de alte entitati, concepute cu scopul de a lega entitatile descrise mai sus intre ele si pentru a forma claritate si logica in baza de date.

CLIENTS(client_id, first_name, last_name, phone_number, city, street, street_number, zip_code)

ORDERS(order_id, *client_id, shop_id, employee_id*, order_status, order_date, shipped_date)

ORDER_DETAILS(order_id, product_id, price, quantity)

PRODUCTS(product_id, *type_id*, product_name, price)

TYPES(type_id, type_name)

STOCKS(shop_id, product_id, quantity)

SHOPS(shop_id, shop_name, city, street, street_number, zip_code)

EMPLOYEES(employee_id**,** *manager_id, shop_id*, first_name, last_name, phone_number)

Diagrama ER:

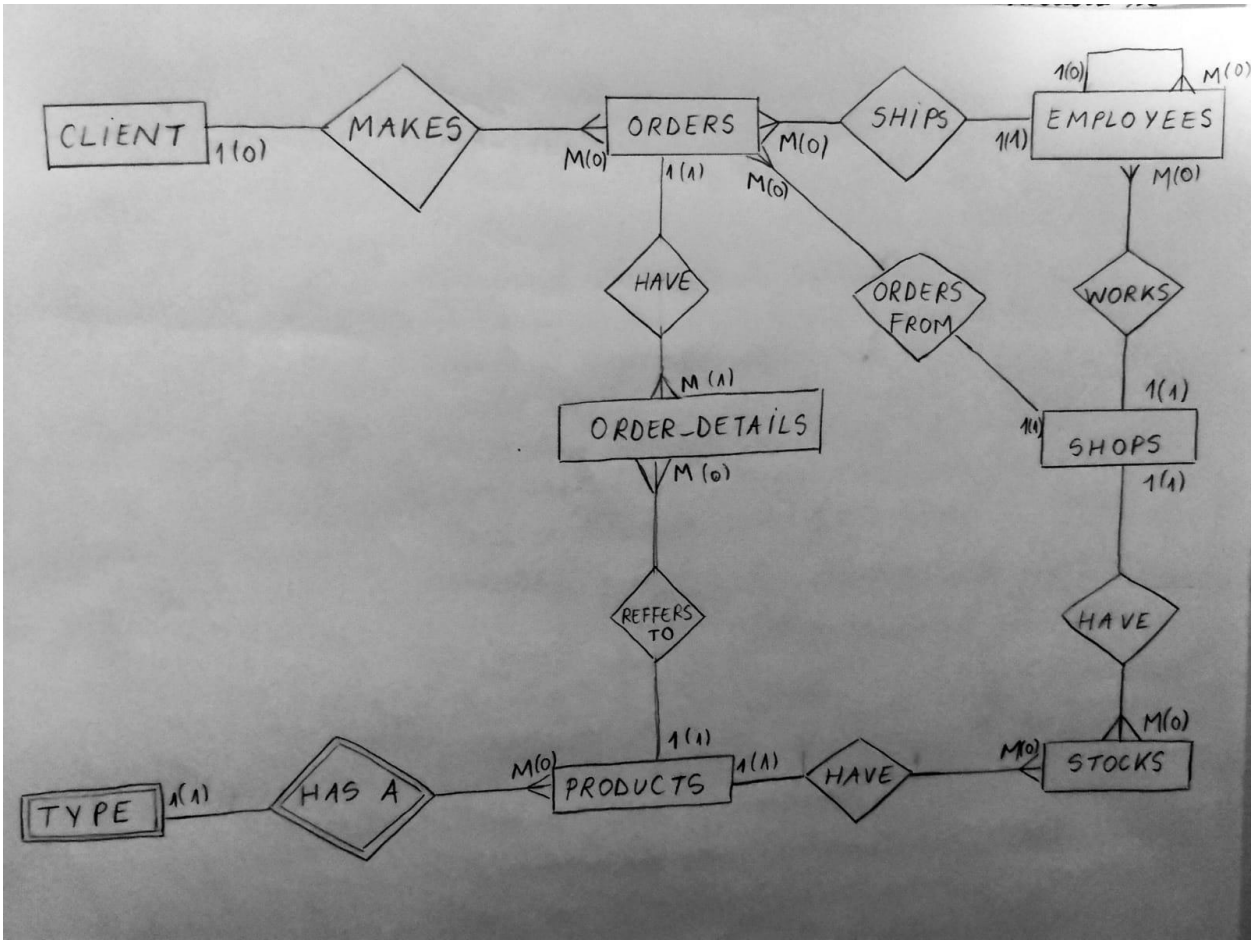
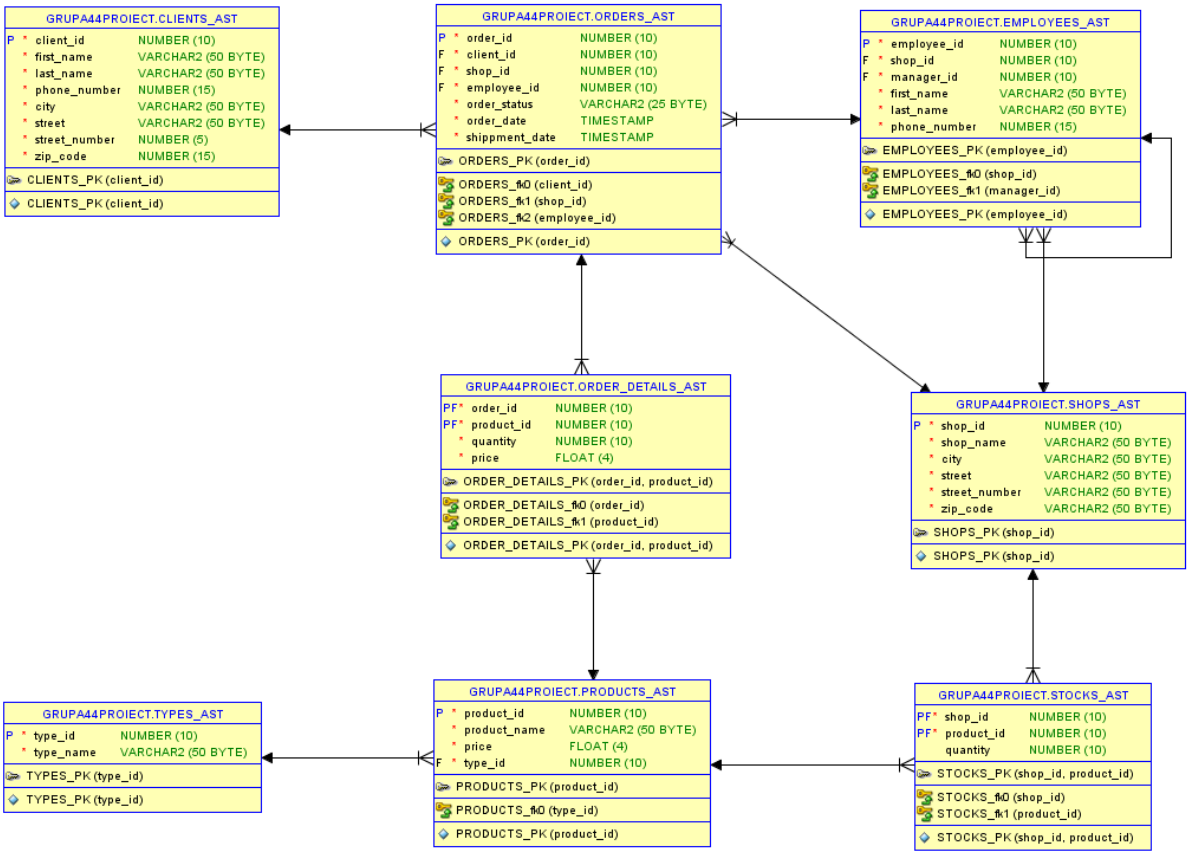

CLIENT
1(0)
MAKES
M(0)
ORDERS
M(0)
SHIPS
1(1)
EMPLOYEES
1(0)
M(0)
M(0)
1(1)
M(0)
HAVE
ORDERS FROM
WORKS
M(1)
1(1)
ORDER_DETAILS
1(1)
SHOPS
M(0)
1(1)
REFFERS TO
HAVE
1(1)
M(0)
TYPE
1(1)
HAS A
M(0)
PRODUCTS
1(1)
HAVE
M(0)
STOCKS

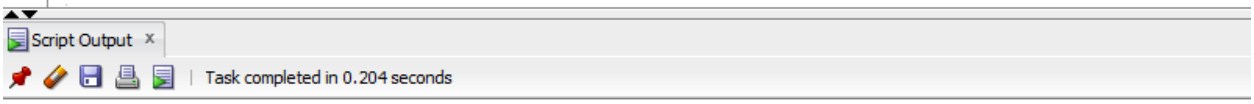
Diagrama conceptuala:



Am facut niste modificari intre timp la “CLIENTS”, “ORDERS” si “EMPLOYEES”, mai exact am schimbat tipul de date pentru numarul de telefon si codul postal, astfel incat sa devina varchar2, iar la “ORDERS” am sters cele doua atribute “shippment_date”, “order_date”.

CREARE TABELE

```
CREATE TABLE "ORDERS_AST" (
    "order_id" NUMBER(10) NOT NULL,
    "client_id" NUMBER(10) NOT NULL,
    "shop_id" NUMBER(10) NOT NULL,
    "employee_id" NUMBER(10) NOT NULL,
    "order_status" VARCHAR2(25) NOT NULL,
    "order_date" TIMESTAMP NOT NULL,
    "shippment_date" TIMESTAMP NOT NULL,
    constraint ORDERS_PK PRIMARY KEY ("order_id"));

CREATE sequence "ORDERS_ORDER_ID_SEQ";

CREATE trigger "ORDERS_ORDER_ID"
  before insert on "ORDERS_AST"
  for each row
begin
  select "ORDERS_ORDER_ID_SEQ".nextval into :NEW."order_id" from dual;
end;
```




```
Sequence ORDERS_ORDER_ID_SEQ dropped.


Table "ORDERS_AST" created.


Sequence "ORDERS_ORDER_ID_SEQ" created.


Trigger ORDERS_ORDER_ID compiled
```

```
CREATE TABLE "ORDERS_AST" (
    "order_id" NUMBER(10) NOT NULL,
    "client_id" NUMBER(10) NOT NULL,
    "shop_id" NUMBER(10) NOT NULL,
    "employee_id" NUMBER(10) NOT NULL,
    "order_status" VARCHAR2(25) NOT NULL,
    "order_date" TIMESTAMP NOT NULL,
    "shippment_date" TIMESTAMP NOT NULL,
    constraint ORDERS_PK PRIMARY KEY ("order_id"));

CREATE sequence "ORDERS_ORDER_ID_SEQ";

ALTER TABLE ORDERS_AST DROP ("shippment_date", "order_date");
```

```
CREATE TABLE "EMPLOYEES_AST" (
    "employee_id" NUMBER(10) NOT NULL,
    "shop_id" NUMBER(10) NOT NULL,
    "manager_id" NUMBER(10) NOT NULL,
    "first_name" VARCHAR2(50) NOT NULL,
    "last_name" VARCHAR2(50) NOT NULL,
    "phone_number" NUMBER(15) NOT NULL,
    constraint EMPLOYEES_PK PRIMARY KEY ("employee_id"));

CREATE sequence "EMPLOYEES_EMPLOYEE_ID_SEQ";

CREATE trigger "EMPLOYEES_EMPLOYEE_ID"
  before insert on "EMPLOYEES_AST"
  for each row
begin
  select "EMPLOYEES_EMPLOYEE_ID_SEQ".nextval into :NEW."employee_id" from dual;
end;
```

Script Output

Task completed in 0.151 seconds

```
Sequence "EMPLOYEES_EMPLOYEE_ID_SEQ" created.


Trigger EMPLOYEES_EMPLOYEE_ID compiled


Table "EMPLOYEES_AST" created.


Sequence "EMPLOYEES_EMPLOYEE_ID_SEQ" created.


Trigger EMPLOYEES_EMPLOYEE_ID compiled
```

Uitasem sa las manager_id sa fie si NULL, am modificat cu ALTER TABLE mai jos.

```
ALTER TABLE EMPLOYEES_AST
MODIFY "manager_id" NULL;

CREATE sequence "EMPLOYEES_EMPLOYEE_ID_SEQ";

CREATE trigger "EMPLOYEES_EMPLOYEE_ID"
  before insert on "EMPLOYEES_AST"
  for each row
begin
  select "EMPLOYEES_EMPLOYEE_ID_SEQ".nextval into :NEW.
```

Script Output ×

Task completed in 0.071 seconds

```
Error starting at line : 29 in command -
ALTER TABLE EMPLOYEES_AST
MODIFY manager_id NULL
Error report -
ORA-00904: "MANAGER_ID": invalid identifier
00904. 00000 -  "%s: invalid identifier"
*Cause:
*Action:

Table EMPLOYEES_AST altered.
```

```
zip_code VARCHAR2(15));

ALTER TABLE EMPLOYEES_AST
MODIFY "phone_number" VARCHAR2(15);

INSERT INTO clients_ast(first_name, last_name, phone_number, city, street, street_number, zip_code)
VALUES ('Ion', 'Popescu', '0769-999-999', 'Bucuresti', 'Florilor', 6, '110165');

CREATE TABLE "TYPES_AST" (
    "type_id" NUMBER(10) NOT NULL,
    "type_name" VARCHAR2(50) NOT NULL,
    constraint TYPES_PK PRIMARY KEY ("type_id"));
```

Script Output ×

Task completed in 11.574 seconds

```
ALTER TABLE "PRODUCTS_AST" ADD CONSTRAINT "PRODUCTS_fk0" FOREIGN KEY ("type_id") REFERENCES "TYPES_AST"("type_id");

ALTER TABLE "STOCKS_AST" ADD CONSTRAINT "STOCKS_fk0" FOREIGN KEY ("shop_id") REFERENCES "SHOPS_AST"("shop_id");
ALTER TABLE "STOCKS_AST" ADD CONSTRAINT "STOCKS_fk1" FOREIGN KEY ("product_id") REFERENCES "PRODUCTS_AST"("product_id");
Error report -
ORA-04081: trigger 'ORDERS_ORDER_ID' already exists
04081. 00000 -  "trigger '%s' already exists"
*Cause:    The TRIGGER name or type already exists.
*Action:   Use a different trigger name or drop the trigger which is of the
           same name.

Table EMPLOYEES_AST altered.
```

```
CREATE TABLE "ORDER_DETAILS_AST" (
    "order_id" NUMBER(10) NOT NULL,
    "product_id" NUMBER(10) NOT NULL,
    "quantity" NUMBER(10) NOT NULL,
    "price" FLOAT(4) NOT NULL,
    constraint ORDER_DETAILS_PK PRIMARY KEY ("order_id","product_id"))

CREATE TABLE "PRODUCTS_AST" (
    "product_id" NUMBER(10) NOT NULL,
    "product_name" VARCHAR2(50) NOT NULL,
    "price" FLOAT(4) NOT NULL,
    "type_id" NUMBER(10) NOT NULL,
    constraint PRODUCTS_PK PRIMARY KEY ("product_id"));

CREATE sequence "PRODUCTS_PRODUCT_ID_SEQ";

CREATE trigger "PRODUCTS_PRODUCT_ID"
```

Script Output

Task completed in 0.069 seconds

```
    "order_id" NUMBER(10) NOT NULL,
    "product_id" NUMBER(10) NOT NULL,
    "quantity" NUMBER(10) NOT NULL,
    "price" FLOAT(4) NOT NULL,
    constraint ORDER_DETAILS_PK PRIMARY KEY ("order_id","product_id"))
Error report -
ORA-00955: name is already used by an existing object
00955. 00000 -  "name is already used by an existing object"
*Cause:
*Action:

Table "ORDER_DETAILS_AST" created.
```

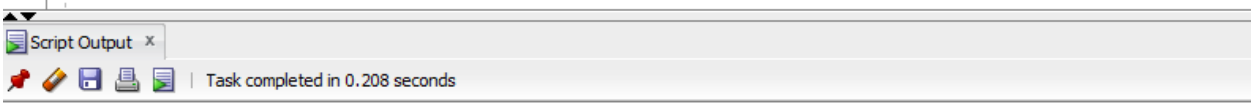
```
    "quantity" NUMBER(10) NOT NULL,
    "price" FLOAT(4) NOT NULL,
    constraint ORDER_DETAILS_PK PRIMARY KEY ("order_id","product_id"));

CREATE TABLE "PRODUCTS_AST" (
    "product_id" NUMBER(10) NOT NULL,
    "product_name" VARCHAR2(50) NOT NULL,
    "price" FLOAT(4) NOT NULL,
    "type_id" NUMBER(10) NOT NULL,
    constraint PRODUCTS_PK PRIMARY KEY ("product_id"));

CREATE sequence "PRODUCTS_PRODUCT_ID_SEQ";

CREATE trigger "PRODUCTS_PRODUCT_ID"
  before insert on "PRODUCTS_AST"
  for each row
begin
  select "PRODUCTS_PRODUCT_ID_SEQ".nextval into :NEW."product_id" from dual;
end;
```

Script Output

Task completed in 0.184 seconds

```
          same name.

Table "ORDER_DETAILS_AST" created.


Table "PRODUCTS_AST" created.


Sequence "PRODUCTS_PRODUCT_ID_SEQ" created.


Trigger PRODUCTS_PRODUCT_ID compiled
```

```
CREATE TABLE "CLIENTS_AST" (
    "client_id" NUMBER(10) NOT NULL,
    "first_name" VARCHAR2(50) NOT NULL,
    "last_name" VARCHAR2(50) NOT NULL,
    "phone_number" NUMBER(15) NOT NULL,
    "city" VARCHAR2(50) NOT NULL,
    "street" VARCHAR2(50) NOT NULL,
    "street_number" NUMBER(5) NOT NULL,
    "zip_code" NUMBER(15) NOT NULL,
    constraint CLIENTS_PK PRIMARY KEY ("client_id"));

CREATE sequence "CLIENTS_CLIENT_ID_SEQ";

CREATE trigger "CLIENTS_CLIENT_ID"
  before insert on "CLIENTS_AST"
  for each row
begin
  select "CLIENTS_CLIENT_ID_SEQ".nextval into :NEW."client_id" from dual;
end;
```




```
Trigger PRODUCTS_PRODUCT_ID compiled


Table "CLIENTS_AST" created.


Sequence "CLIENTS_CLIENT_ID_SEQ" created.


Trigger CLIENTS_CLIENT_ID compiled
```

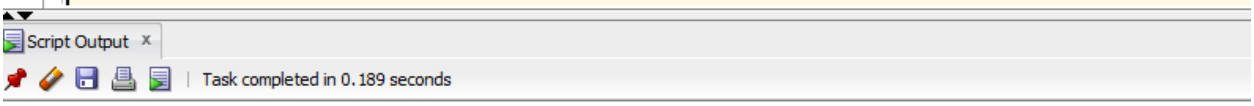
```
    "city" VARCHAR2(30) NOT NULL,
    "street" VARCHAR2(50) NOT NULL,
    "street_number" NUMBER(5) NOT NULL,
    "zip_code" NUMBER(15) NOT NULL,
    constraint CLIENTS_PK PRIMARY KEY ("client_id"));

CREATE sequence "CLIENTS_CLIENT_ID_SEQ";

CREATE trigger "CLIENTS_CLIENT_ID"
  before insert on "CLIENTS_AST"
  for each row
begin
  select "CLIENTS_CLIENT_ID_SEQ".nextval into :NEW."client_id" from dual;
end;

ALTER TABLE CLIENTS_AST
  MODIFY ("phone_number" VARCHAR2(15),
            "zip_code" VARCHAR2(15));

INSERT INTO clients_ast(first_name, last_name, phone_number, city, street, street_number, zip_code)
VALUES ('Ion', 'Popescu', 2769999999, 'Bucuresti', 'Florilor', 6, 11016);
```

Script Output ×

Task completed in 0.088 seconds

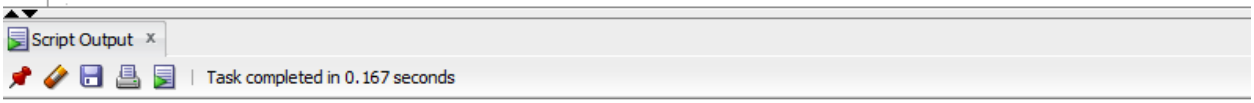
```
Error starting at line : 88 in command -
ALTER TABLE CLIENTS_AST
  MODIFY ("phone_number" VARCHAR2(15) NOT NULL,
            "zip_code" VARCHAR2(15) NOT NULL)
Error report -
ORA-01442: column to be modified to NOT NULL is already NOT NULL
01442. 00000 -  "column to be modified to NOT NULL is already NOT NULL"
*Cause:
*Action:

Table CLIENTS_AST altered.
```

```
  before insert on "CLIENTS_AST"
  for each row
begin
  select "CLIENTS_CLIENT_ID_SEQ".nextval into :NEW."client_id" from dual;
end;

CREATE TABLE "TYPES_AST" (
    "type_id" NUMBER(10) NOT NULL,
    "type_name" VARCHAR2(50) NOT NULL,
    constraint TYPES_PK PRIMARY KEY ("type_id"));

CREATE sequence "TYPES_TYPE_ID_SEQ";

CREATE trigger "TYPES_TYPE_ID"
  before insert on "TYPES_AST"
  for each row
begin
  select "TYPES_TYPE_ID_SEQ".nextval into :NEW."type_id" from dual;
end;
```




```
Trigger CLIENTS_CLIENT_ID compiled


Table "TYPES_AST" created.


Sequence "TYPES_TYPE_ID_SEQ" created.


Trigger TYPES_TYPE_ID compiled
```

```
  select "TYPES_TYPE_ID_SEQ".nextval into :NEW."type_id" from dual;
end;

CREATE TABLE "SHOPS_AST" (
    "shop_id" NUMBER(10) NOT NULL,
    "shop_name" VARCHAR2(50) NOT NULL,
    "city" VARCHAR2(50) NOT NULL,
    "street" VARCHAR2(50) NOT NULL,
    "street_number" VARCHAR2(50) NOT NULL,
    "zip_code" VARCHAR2(50) NOT NULL,
    constraint SHOPS_PK PRIMARY KEY ("shop_id"));

CREATE sequence "SHOPS_SHOP_ID_SEQ";

CREATE trigger "SHOPS_SHOP_ID"
  before insert on "SHOPS_AST"
  for each row
begin
  select "SHOPS_SHOP_ID_SEQ".nextval into :NEW."shop_id" from dual;
end;
```




```
Trigger TYPES_TYPE_ID compiled


Table "SHOPS_AST" created.


Sequence "SHOPS_SHOP_ID_SEQ" created.


Trigger SHOPS_SHOP_ID compiled
```

```
CREATE sequence "SHOPS_SHOP_ID_SEQ";

CREATE trigger "SHOPS_SHOP_ID"
  before insert on "SHOPS_AST"
  for each row
begin
  select "SHOPS_SHOP_ID_SEQ".nextval into :NEW."shop_id" from dual;
end;

CREATE TABLE "STOCKS_AST" (
    "shop_id" NUMBER(10) NOT NULL,
    "product_id" NUMBER(10) NOT NULL,
    "quantity" NUMBER(10),
    constraint STOCKS_PK PRIMARY KEY ("shop_id","product_id"));


ALTER TABLE "ORDERS_AST" ADD CONSTRAINT "ORDERS_fk0" FOREIGN KEY ("client_id") REFERENCES "CLIENTS_AST"("client_id");
ALTER TABLE "ORDERS_AST" ADD CONSTRAINT "ORDERS_fk1" FOREIGN KEY ("shop_id") REFERENCES "SHOPS_AST"("shop_id");
ALTER TABLE "ORDERS_AST" ADD CONSTRAINT "ORDERS_fk2" FOREIGN KEY ("employee_id") REFERENCES "EMPLOYEES_AST"("employee_id
```

Script Output

Task completed in 0.075 seconds

```
*Cause:
*Action:

Error starting at line : 140 in command -
ALTER TABLE "STOCKS_AST" ADD CONSTRAINT "STOCKS_fk1" FOREIGN KEY ("product_id") REFERENCES "PRODUCTS_AST"("product_id")
Error report -
ORA-00942: table or view does not exist
00942. 00000 -  "table or view does not exist"
*Cause:
*Action:

Table "STOCKS_AST" created.
```

```
    "quantity" NUMBER(10),
    constraint STOCKS_PK PRIMARY KEY ("shop_id","product_id"));


ALTER TABLE "ORDERS_AST" ADD CONSTRAINT "ORDERS_fk0" FOREIGN KEY ("client_id") REFERENCES "CLIENTS_AST"("client_id");
ALTER TABLE "ORDERS_AST" ADD CONSTRAINT "ORDERS_fk1" FOREIGN KEY ("shop_id") REFERENCES "SHOPS_AST"("shop_id");
ALTER TABLE "ORDERS_AST" ADD CONSTRAINT "ORDERS_fk2" FOREIGN KEY ("employee_id") REFERENCES "EMPLOYEES_AST"("employee_id");

ALTER TABLE "EMPLOYEES_AST" ADD CONSTRAINT "EMPLOYEES_fk0" FOREIGN KEY ("shop_id") REFERENCES "SHOPS_AST"("shop_id");
ALTER TABLE "EMPLOYEES_AST" ADD CONSTRAINT "EMPLOYEES_fk1" FOREIGN KEY ("manager_id") REFERENCES "EMPLOYEES_AST"("employee_id");

ALTER TABLE "ORDER_DETAILS_AST" ADD CONSTRAINT "ORDER_DETAILS_fk0" FOREIGN KEY ("order_id") REFERENCES "ORDERS_AST"("order_id");
ALTER TABLE "ORDER_DETAILS_AST" ADD CONSTRAINT "ORDER_DETAILS_fk1" FOREIGN KEY ("product_id") REFERENCES "PRODUCTS_AST"("product_id");

ALTER TABLE "PRODUCTS_AST" ADD CONSTRAINT "PRODUCTS_fk0" FOREIGN KEY ("type_id") REFERENCES "TYPES_AST"("type_id");

ALTER TABLE "STOCKS_AST" ADD CONSTRAINT "STOCKS_fk0" FOREIGN KEY ("shop_id") REFERENCES "SHOPS_AST"("shop_id");
ALTER TABLE "STOCKS_AST" ADD CONSTRAINT "STOCKS_fk1" FOREIGN KEY ("product_id") REFERENCES "PRODUCTS_AST"("product_id");
```

Script Output

Task completed in 0.253 seconds

```
Table "ORDERS_AST" altered.


Table "ORDERS_AST" altered.


Table "EMPLOYEES_AST" altered.


Table "EMPLOYEES_AST" altered.
```



```
    "quantity" NUMBER(10),
    constraint STOCKS_PK PRIMARY KEY ("shop_id","product_id"));


ALTER TABLE "ORDERS_AST" ADD CONSTRAINT "ORDERS_fk0" FOREIGN KEY ("client_id") REFERENCES "CLIENTS_AST"("client_id");
ALTER TABLE "ORDERS_AST" ADD CONSTRAINT "ORDERS_fk1" FOREIGN KEY ("shop_id") REFERENCES "SHOPS_AST"("shop_id");
ALTER TABLE "ORDERS_AST" ADD CONSTRAINT "ORDERS_fk2" FOREIGN KEY ("employee_id") REFERENCES "EMPLOYEES_AST"("employee_id");

ALTER TABLE "EMPLOYEES_AST" ADD CONSTRAINT "EMPLOYEES_fk0" FOREIGN KEY ("shop_id") REFERENCES "SHOPS_AST"("shop_id");
ALTER TABLE "EMPLOYEES_AST" ADD CONSTRAINT "EMPLOYEES_fk1" FOREIGN KEY ("manager_id") REFERENCES "EMPLOYEES_AST"("employee_id");

ALTER TABLE "ORDER_DETAILS_AST" ADD CONSTRAINT "ORDER_DETAILS_fk0" FOREIGN KEY ("order_id") REFERENCES "ORDERS_AST"("order_id");
ALTER TABLE "ORDER_DETAILS_AST" ADD CONSTRAINT "ORDER_DETAILS_fk1" FOREIGN KEY ("product_id") REFERENCES "PRODUCTS_AST"("product_id");

ALTER TABLE "PRODUCTS_AST" ADD CONSTRAINT "PRODUCTS_fk0" FOREIGN KEY ("type_id") REFERENCES "TYPES_AST"("type_id");

ALTER TABLE "STOCKS_AST" ADD CONSTRAINT "STOCKS_fk0" FOREIGN KEY ("shop_id") REFERENCES "SHOPS_AST"("shop_id");
ALTER TABLE "STOCKS_AST" ADD CONSTRAINT "STOCKS_fk1" FOREIGN KEY ("product_id") REFERENCES "PRODUCTS_AST"("product_id");
```

Script Output

Task completed in 0.222 seconds

```
Table "EMPLOYEES_AST" altered.


Table "ORDER_DETAILS_AST" altered.


Table "ORDER_DETAILS_AST" altered.


Table "PRODUCTS_AST" altered.
```

```
    "quantity" NUMBER(10),
    constraint STOCKS_PK PRIMARY KEY ("shop_id","product_id"));


ALTER TABLE "ORDERS_AST" ADD CONSTRAINT "ORDERS_fk0" FOREIGN KEY ("client_id") REFERENCES "CLIENTS_AST"("client_id");
ALTER TABLE "ORDERS_AST" ADD CONSTRAINT "ORDERS_fk1" FOREIGN KEY ("shop_id") REFERENCES "SHOPS_AST"("shop_id");
ALTER TABLE "ORDERS_AST" ADD CONSTRAINT "ORDERS_fk2" FOREIGN KEY ("employee_id") REFERENCES "EMPLOYEES_AST"("employee_id");

ALTER TABLE "EMPLOYEES_AST" ADD CONSTRAINT "EMPLOYEES_fk0" FOREIGN KEY ("shop_id") REFERENCES "SHOPS_AST"("shop_id");
ALTER TABLE "EMPLOYEES_AST" ADD CONSTRAINT "EMPLOYEES_fk1" FOREIGN KEY ("manager_id") REFERENCES "EMPLOYEES_AST"("employee_id");

ALTER TABLE "ORDER_DETAILS_AST" ADD CONSTRAINT "ORDER_DETAILS_fk0" FOREIGN KEY ("order_id") REFERENCES "ORDERS_AST"("order_id");
ALTER TABLE "ORDER_DETAILS_AST" ADD CONSTRAINT "ORDER_DETAILS_fk1" FOREIGN KEY ("product_id") REFERENCES "PRODUCTS_AST"("product_id");

ALTER TABLE "PRODUCTS_AST" ADD CONSTRAINT "PRODUCTS_fk0" FOREIGN KEY ("type_id") REFERENCES "TYPES_AST"("type_id");

ALTER TABLE "STOCKS_AST" ADD CONSTRAINT "STOCKS_fk0" FOREIGN KEY ("shop_id") REFERENCES "SHOPS_AST"("shop_id");
ALTER TABLE "STOCKS_AST" ADD CONSTRAINT "STOCKS_fk1" FOREIGN KEY ("product_id") REFERENCES "PRODUCTS_AST"("product_id");
```

Script Output | Task completed in 0.222 seconds

```
Error starting at line : 137 in command -
ALTER TABLE "PRODUCTS_AST" ADD CONSTRAINT "PRODUCTS_fk0" FOREIGN KEY ("type_id") REFERENCES "TYPES_AST"("type_id")
Error report -
ORA-02275: such a referential constraint already exists in the table
02275. 00000 -  "such a referential constraint already exists in the table"
*Cause:    Self-evident.
*Action:   Remove the extra constraint.

Table "STOCKS_AST" altered.


Table "STOCKS_AST" altered.
```


(Selectasem prea mult cod)

```
ALTER TABLE EMPLOYEES_AST
MODIFY "phone_number" VARCHAR2(15);

INSERT INTO clients_ast("first_name", "last_name", "phone_number", "city", "street", "street_number", "zip_code")
VALUES ('Ion', 'Popescu', '0769-999-999', 'Bucuresti', 'Florilor', 6, '110165');
INSERT INTO clients_ast("first_name", "last_name", "phone_number", "city", "street", "street_number", "zip_code")
VALUES ('Mircea', 'Radulescu', '0769-999-998', 'Bucuresti', 'Ciresilor', 1, '022452');
INSERT INTO clients_ast("first_name", "last_name", "phone_number", "city", "street", "street_number", "zip_code")
VALUES ('Anca', 'Petrov', '0769-999-997', 'Cluj', 'Rozelor', 10, '113265');

CREATE TABLE "TYPES_AST" (
    "type_id" NUMBER(10) NOT NULL,
    "type_name" VARCHAR2(50) NOT NULL
```

Script Output | Task completed in 0.592 seconds

```
           same name.

Table EMPLOYEES_AST altered.


1 row inserted.


1 row inserted.


1 row inserted.
```

```
SELECT *
FROM clients_ast;
```

Script Output | Query Result

SQL | All Rows Fetched: 3 in 10.74 seconds

	client_id	first_name	last_name	phone_number	city	street	street_number	zip_code
1	1	Ion	Popescu	0769-99...	Bucu...	Flor...	6	110165
2	2	Mircea	Radu...	0769-99...	Bucu...	Cire...	1	022452
3	3	Anca	Petrov	0769-99...	Cluj	Rozelor	10	113265

```
INSERT INTO EMPLOYEES_AST("manager_id","shop_id","first_name","last_name","phone_number")
VALUES(Null,1,'Robert','Cristea','0768-967-012');
INSERT INTO EMPLOYEES_AST("manager_id","shop_id","first_name","last_name","phone_number")
VALUES(1,1,'Rares','Neagu','0768-961-012');
INSERT INTO EMPLOYEES_AST("manager_id","shop_id","first_name","last_name","phone_number")
VALUES(Null,2,'Marin','Marian','0768-967-142');
INSERT INTO EMPLOYEES_AST("manager_id","shop_id","first_name","last_name","phone_number")
VALUES(3,2,'Andrei','Popa','0768-967-012');
INSERT INTO EMPLOYEES_AST("manager_id","shop_id","first_name","last_name","phone_number")
VALUES(Null,3,'Matei','Cosmin','0768-967-992');
INSERT INTO EMPLOYEES_AST("manager_id","shop_id","first_name","last_name","phone_number")
VALUES(5,3,'Miruna','Irinel','0768-967-432');


INSERT INTO ORDERS_AST("client_id","shop_id","employee_id","order_status","order_date","order_time")
```

Script Output | Task completed in 0.241 seconds

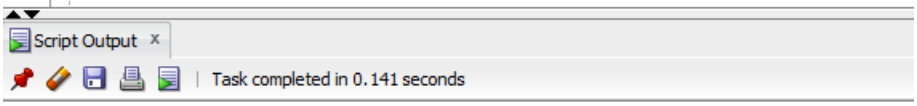
```
1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.
```

```
SELECT *
FROM employees_ast;

SELECT *
FROM orders_ast;
```

Script Output | Query Result | SQL | All Rows Fetched: 6 in 0.011 seconds

	employee_id	shop_id	manager_id	first_name	last_name	phone_number
1	1	1	(null)	Robert	Cristea	0768-967-012
2	2	1	1	Rares	Neagu	0768-961-012
3	3	2	(null)	Marin	Marian	0768-967-142
4	4	2	3	Andrei	Popa	0768-967-012
5	5	3	(null)	Matei	Cosmin	0768-967-992
6	6	3	5	Miruna	Irinel	0768-967-432

```
INSERT INTO TYPES_AST("type_name")
VALUES ('tablou');
INSERT INTO TYPES_AST("type_name")
VALUES ('figurina');
INSERT INTO TYPES_AST("type_name")
VALUES ('comic book');

INSERT INTO PRODUCTS_AST("type_id","product_name","pr
VALUES(5,'Peisaj de iarna', 100);
INSERT INTO PRODUCTS_AST("type_id","product_name","pr
VALUES(5,'Ilustratie Void Chicken', 60);
```




```
1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.
```

```
SELECT *
FROM types_ast;

SELECT *
FROM products_ast;

SELECT *
FROM shops_ast;
```

Script Output | Query Result

SQL | All Rows Fetched: 3 in 0.01 seconds

	type_id	type_name
1	1	tablou
2	2	figurina
3	3	comic book

```
INSERT INTO PRODUCTS_AST("type_id","product_name","price")
VALUES(1,'Peisaj de iarna', 100);
INSERT INTO PRODUCTS_AST("type_id","product_name","price")
VALUES(1,'Ilustratie Void Chicken', 60);
INSERT INTO PRODUCTS_AST("type_id","product_name","price")
VALUES(2,'Figurina Lux Lyfeld', 110);
INSERT INTO PRODUCTS_AST("type_id","product_name","price")
VALUES(3,'Lux si Nox Volumul 1', 50);

INSERT INTO SHOPS_AST("shop_name","city","street","street_number","zip_code")
VALUES('Artistique','Bucuresti','Roua',21,'0713435');
INSERT INTO SHOPS_AST("shop_name","city","street","street_number","zip_code")
VALUES('Beart','Bucuresti','Laleaua',3,'02142435');
```

Script Output

Task completed in 0.135 seconds

```
1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.
```

```sql
SELECT *
FROM products_ast;

SELECT *
FROM shops_ast;

SELECT *
FROM employees_ast;

SELECT *
FROM orders_ast;
```

Script Output | Query Result

SQL | All Rows Fetched: 4 in 0.011 seconds

	product_id	product_name	price	type_id
1	1	Peisaj de iarna	100	1
2	2	Ilustratie Void Chicken	60	1
3	3	Figurina Lux Lyfeld	110	2
4	4	Lux si Nox Volumul 1	50	3

```sql
SELECT *
FROM products_ast;

SELECT *
FROM shops_ast;

SELECT *
FROM employees_ast;

SELECT *
FROM orders_ast;
```

Script Output | Query Result

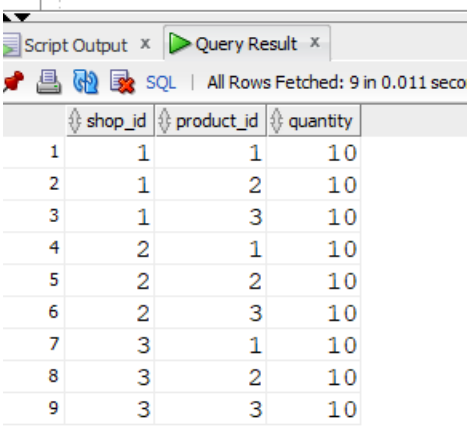
SQL | All Rows Fetched: 3 in 0.012 seconds

	shop_id	shop_name	city	street	street_number	zip_code
1	1	Artistique	Bucuresti	Roua	21	0713435
2	2	Beart	Bucuresti	Laleaua	3	02142435
3	3	Bookify	Bucuresti	Roua	15	07138563

```
INSERT INTO STOCKS_AST("shop_id", "product_id", "quantity")
VALUES(1,1,10);
INSERT INTO STOCKS_AST("shop_id", "product_id", "quantity")
VALUES(1,2,10);
INSERT INTO STOCKS_AST("shop_id", "product_id", "quantity")
VALUES(1,3,10);
INSERT INTO STOCKS_AST("shop_id", "product_id", "quantity")
VALUES(2,1,10);
INSERT INTO STOCKS_AST("shop_id", "product_id", "quantity")
VALUES(2,2,10);
INSERT INTO STOCKS_AST("shop_id", "product_id", "quantity")
VALUES(2,3,10);
INSERT INTO STOCKS_AST("shop_id", "product_id", "quantity")
VALUES(3,1,10);
INSERT INTO STOCKS_AST("shop_id", "product_id", "quantity")
VALUES(3,2,10);
INSERT INTO STOCKS_AST("shop_id", "product_id", "quantity")
VALUES(3,3,10);
```


```
1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.
```

```
SELECT *
FROM stocks_ast;
```



	shop_id	product_id	quantity
1	1	1	10
2	1	2	10
3	1	3	10
4	2	1	10
5	2	2	10
6	2	3	10
7	3	1	10
8	3	2	10
9	3	3	10

```
INSERT INTO ORDERS_AST("client_id","shop_id","employee_id","order_status")
VALUES(1,1,1,'shipped');
INSERT INTO ORDERS_AST("client_id","shop_id","employee_id","order_status")
VALUES(1,1,2,'arrived');
INSERT INTO ORDERS_AST("client_id","shop_id","employee_id","order_status")
VALUES(1,1,2,'arrived');
INSERT INTO ORDERS_AST("client_id","shop_id","employee_id","order_status")
VALUES(1,3,5,'arrived');
INSERT INTO ORDERS_AST("client_id","shop_id","employee_id","order_status")
VALUES(1,3,6,'arrived');
INSERT INTO ORDERS_AST("client_id","shop_id","employee_id","order_status")
VALUES(2,1,1,'shipped');
INSERT INTO ORDERS_AST("client_id","shop_id","employee_id","order_status")
VALUES(3,1,2,'arrived');
INSERT INTO ORDERS_AST("client_id","shop_id","employee_id","order_status")
VALUES(3,3,5,'shipped');
INSERT INTO ORDERS_AST("client_id","shop_id","employee_id","order_status")
VALUES(3,3,6,'arrived');
INSERT INTO ORDERS_AST("client_id","shop_id","employee_id","order_status")
VALUES(3,2,1,'arrived');
INSERT INTO ORDERS_AST("client_id","shop_id","employee_id","order_status")
VALUES(2,1,2,'arrived');
```

Script Output

Task completed in 0.257 seconds

```
1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.
```

```
SELECT *
FROM orders_ast;

SELECT *
FROM stocks_ast;
```

Script Output | Query Result

SQL | All Rows Fetched: 11 in 0.009 seconds

	order_id	client_id	shop_id	employee_id	order_sta...
1	1	1	1	1	shipped
2	2	1	1	2	arrived
3	3	1	1	2	arrived
4	4	1	3	5	arrived
5	5	1	3	6	arrived
6	6	2	1	1	shipped
7	7	3	1	2	arrived
8	8	3	3	5	shipped
9	9	3	3	6	arrived
10	10	3	2	1	arrived
11	11	2	1	2	arrived

```
INSERT INTO ORDER_DETAILS_AST("order_id", "product_id", "price", "quantity")
VALUES(1,1,100,1);
INSERT INTO ORDER_DETAILS_AST("order_id", "product_id", "price", "quantity")
VALUES(2,1,300,3);
INSERT INTO ORDER_DETAILS_AST("order_id", "product_id", "price", "quantity")
VALUES(3,3,50,1);
INSERT INTO ORDER_DETAILS_AST("order_id", "product_id", "price", "quantity")
VALUES(4,3,100,2);
INSERT INTO ORDER_DETAILS_AST("order_id", "product_id", "price", "quantity")
VALUES(5,1,60,2);
INSERT INTO ORDER_DETAILS_AST("order_id", "product_id", "price", "quantity")
VALUES(6,3,110,1);
INSERT INTO ORDER_DETAILS_AST("order_id", "product_id", "price", "quantity")
VALUES(7,3,550,5);
INSERT INTO ORDER_DETAILS_AST("order_id", "product_id", "price", "quantity")
VALUES(8,1,200,2);
INSERT INTO ORDER_DETAILS_AST("order_id", "product_id", "price", "quantity")
VALUES(9,1,100,1);
INSERT INTO ORDER_DETAILS_AST("order_id", "product_id", "price", "quantity")
VALUES(10,3,110,1);
INSERT INTO ORDER_DETAILS_AST("order_id", "product_id", "price", "quantity")
VALUES(11,4,50,1);
```

Script Output

Task completed in 0.303 seconds

```
1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.
```

```sql
SELECT *
FROM order_details_ast;
```

Script Output | Query Result

SQL | All Rows Fetched: 11 in 0.011 seconds

	order_id	product_id	quantity	price
1	1	1	1	100
2	2	1	3	300
3	3	3	1	50
4	4	3	2	100
5	5	1	2	60
6	6	3	1	110
7	7	3	5	550
8	8	1	2	200
9	9	1	1	100
10	10	3	1	110
11	11	4	1	50

INTEROGARI

```sql
-- Numarul de comenzi ale fiecarui client, sortate in functie de id-ul clientului

SELECT "client_id", COUNT("order_id")
FROM orders_ast
GROUP BY "client_id"
ORDER BY "client_id";
```

Query Result

SQL | All Rows Fetched: 3 in 0.01 seconds

	client_id	COUNT("ORDER_ID")
1	1	5
2	2	2
3	3	4

```sql
-- 2. Gaseste comenzile cu valoare mai mare decat 100 al caror id al produselor este intre 1 si 3
SELECT "order_id", COUNT("product_id") product_count, SUM("price") total_price
FROM order_details_ast
GROUP BY "order_id"
HAVING COUNT("product_id") BETWEEN 1 AND 3 AND SUM("price") > 100
ORDER BY total_price DESC, product_count DESC;
```

Query Result

SQL | All Rows Fetched: 5 in 0.01 seconds

	order_id	PRODUCT_COUNT	TOTAL_PRICE
1	7	1	550
2	2	1	300
3	8	1	200
4	6	1	110
5	10	1	110

```sql
-- 3. Selecteaza angajatii care au numele de familie Cristea
SELECT "first_name", "last_name"
FROM employees_ast
WHERE LOWER("last_name") = 'cristea';
```

Query Result

SQL | All Rows Fetched: 1 in 0.011 seconds

	first_name	last_name
1	Robert	Cristea

```sql
-- 4. Selecteaza angajatii si afiseaza numarul lor grupati dupa primele doua litere prenumelui
SELECT SUBSTR("first_name", 1, 2) first_two_letters, COUNT(*)
FROM employees_ast
GROUP BY SUBSTR("first_name", 1, 2)
ORDER BY first_two_letters;
```

Query Result

SQL | All Rows Fetched: 5 in 0.012 seconds

	FIRST_TWO_LETTERS	COUNT(*)
1	An	1
2	Ma	2
3	Mi	1
4	Ra	1
5	Ro	1

```sql
-- 5. Returneaza statusul comenzii si id-ul comenzii asignate fiecarui angajat (angajatii se ocupa de livrari)
SELECT "order_id", "order_status", "first_name", "last_name"
FROM orders_ast o
LEFT JOIN employees_ast e
ON e."employee_id" = o."employee_id";
```

Query Result

SQL | All Rows Fetched: 11 in 0.01 seconds

	order_id	order_status	first_name	last_name
1	10	arrived	Robert	Cristea
2	6	shipped	Robert	Cristea
3	1	shipped	Robert	Cristea
4	11	arrived	Rares	Neagu
5	7	arrived	Rares	Neagu
6	3	arrived	Rares	Neagu
7	2	arrived	Rares	Neagu
8	8	shipped	Matei	Cosmin
9	4	arrived	Matei	Cosmin
10	9	arrived	Miruna	Irinel
11	5	arrived	Miruna	Irinel

```sql
-- 6. Afiseaza angajatii cu seful mentionat
SELECT "employee_id", COALESCE(TO_CHAR(NULLIF("manager_id","employee_id")), 'His own boss') boss
FROM employees_ast;
```

Query Result

SQL | All Rows Fetched: 6 in 0.009 seconds

	employee_id	BOSS
1	1	His own boss
2	2	1
3	3	His own boss
4	4	3
5	5	His own boss
6	6	5

```sql
-- 8. Categoria de produse a carui cel mai mic pret de produs este mai mare de 50
SELECT "type_name", MIN("price") pret_peste_50_lei
FROM products_ast
INNER JOIN types_ast USING("type_id")
GROUP BY "type_name"
HAVING MIN("price") > 50
ORDER BY "type_name";
```

Query Result

SQL | All Rows Fetched: 2 in 0.044 seconds

	type_name	PRET_PESTE_50_LEI
1	figurina	110
2	tablou	60

```sql
-- 7. Afiseaza ierarhia managerului Cristea
SELECT LEVEL, "employee_id", "last_name", "manager_id"
FROM employees_ast
START WITH "employee_id" = (SELECT "employee_id" FROM employees_ast WHERE UPPER("last_name")
LIKE 'CRISTEA')
CONNECT BY PRIOR  "employee_id" = "manager_id";

-- 8. Categoria de produse a carui cel mai mic pret de produs este mai mare de 50
SELECT "type_name", MIN("price") pret_peste_50_lei
FROM products_ast
INNER JOIN types_ast USING("type_id")
GROUP BY "type_name"
```

Query Result

SQL | All Rows Fetched: 2 in 0.01 seconds

	LEVEL	employee_id	last_name	manager_id
1	1	1	Cristea	(null)
2	2	2	Neagu	1

```sql
-- 9. Subinterogarea intoarce media preturilor dupa categoria de produs
-- iar interogarea returneaza media mediilor subinterogarii
SELECT ROUND(AVG(avg_price), 2) avg_of_avg
FROM(SELECT AVG("price") avg_price
    FROM products_ast
    GROUP BY "type_id"
);
```

Query Result

SQL | All Rows Fetched: 1 in 0.012 seconds

	AVG_OF_AVG
1	80

```sql
-- 10. Selecteaza categoria de produse si pretul pentru cele mai scumpe produse per fiecare categorie
-- Returneaza doar produsul al carui cel mai mare pret este intre 50 si 100
SELECT "type_name", MAX("price")
FROM products_ast
INNER JOIN types_ast
USING("type_id")
GROUP BY "type_name"
HAVING MAX("price") BETWEEN 50 AND 100
ORDER BY "type_name";
```

Query Result

SQL | All Rows Fetched: 2 in 0.01 seconds

	type_name	MAX("PRICE")
1	comic book	50
2	tablou	100

```sql
-- 12. Selecteaza toti angajatii si vanzarile lor
SELECT DISTINCT "first_name", "last_name", "order_id", "order_status"
FROM orders_ast o
RIGHT JOIN employees_ast e ON e."employee_id" = o."employee_id"
ORDER BY "first_name", "last_name";

-- 13.

-- 14.

-- 15.
```

Query Result

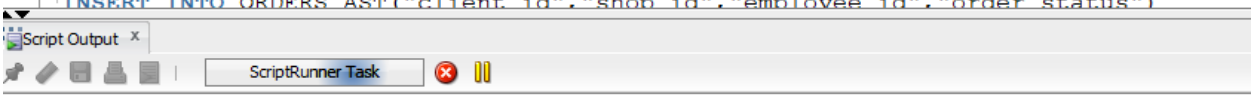
SQL | All Rows Fetched: 13 in 0.015 seconds

	first_name	last_name	order_id	order_status
1	Andrei	Popa	(null)	(null)
2	Marin	Marian	(null)	(null)
3	Matei	Cosmin	15	arrived
4	Matei	Cosmin	19	shipped
5	Miruna	Irinel	16	arrived
6	Miruna	Irinel	20	arrived
7	Rares	Neagu	13	arrived
8	Rares	Neagu	14	arrived
9	Rares	Neagu	18	arrived
10	Rares	Neagu	22	arrived
11	Robert	Cristea	12	shipped
12	Robert	Cristea	17	shipped
13	Robert	Cristea	21	arrived

Cineva mi-a sters datele din tabelele orders, stocks si order_details si ar fi trebuit sa le refac, dar orders a luat-o de la un alt index (nu am mai resetat) si order_details nu se mai potriveste si iar mi s-au blocat lucrurile.

Am incercat sa remediez indecsii ca sa pot reface tabelele, insa acum imi ruleaza la infinit (de data asta nu merge nici daca inchid si redeschid sql-urile).

```
INSERT INTO ORDERS_AST("client_id","shop_id","employee_id","order_status")
VALUES(1,1,1,'shipped');
INSERT INTO ORDERS_AST("client_id","shop_id","employee_id","order_status")
VALUES(1,1,2,'arrived');
INSERT INTO ORDERS_AST("client_id","shop_id","employee_id","order_status")
```



Acum sa speram ca voi reusi sa fac restul interogarilor ramase doar cu tabelele pe care le am disponibile.

```
INSERT INTO STOCKS_AST("shop_id", "product_id", "quantity")
VALUES(1,1,10);
INSERT INTO STOCKS_AST("shop_id", "product_id", "quantity")
VALUES(1,2,10);
INSERT INTO STOCKS_AST("shop_id", "product_id", "quantity")
VALUES(1,3,10);
INSERT INTO STOCKS_AST("shop_id", "product_id", "quantity")
VALUES(2,1,10);
INSERT INTO STOCKS_AST("shop_id", "product_id", "quantity")
VALUES(2,2,10);
INSERT INTO STOCKS_AST("shop_id", "product_id", "quantity")
```

La stocuri patesc la fel.

```sql
-- 11. Angajatii care nu participa la livrari si nu sunt manageri
SELECT "first_name", "last_name"
FROM employees_ast e
FULL OUTER JOIN orders_ast o ON e."employee_id" = o."employee_id"
WHERE "manager_id" IS NULL
ORDER BY "first_name", "last_name";

-- 12. Selecteaza toti angajatii si vanzarile lor
SELECT DISTINCT "first_name", "last_name", "order_id", "order_status"
FROM orders_ast o
```

Query Result

SQL | All Rows Fetched: 3 in 0.009 seconds

	first_name	last_name
1	Marin	Marian
2	Matei	Cosmin
3	Robert	Cristea

```sql
-- 13. Selecteaza produsele care au pretul mai mare decat media aritmetica a preturilor
-- tuturor produselor
SELECT "product_id", "product_name", "price"
FROM products_ast
WHERE "price" > (SELECT AVG("price")
                 FROM products_ast)
ORDER BY "price";

-- 14.
```

Query Result

SQL | All Rows Fetched: 2 in 0.033 seconds

	product_id	product_name	price
1	1	Peisaj de iarna	100
2	3	Figurina Lux Lyfeld	110

```sql
-- 14. Selecteaza codul magazinului, numele magazinului si numarul de angajati care lucreaza
-- in acel magazin pentru magazinul cu numar maxim de angajati
SELECT "shop_id", "shop_name", COUNT("employee_id") numar_angajati
FROM employees_ast
JOIN shops_ast USING("shop_id")
GROUP BY "shop_id", "shop_name"
HAVING COUNT("employee_id") = (SELECT MAX(COUNT("employee_id"))
                               FROM employees_ast
                               GROUP BY "shop_id");
```

Query Result

SQL | All Rows Fetched: 1 in 0.013 seconds

	shop_id	shop_name	NUMAR_ANGAJATI
1	3	Bookify	3

```sql
-- 16. Numele angajatilor care au lucrat in cel putin aceleasi magazine ca si angajatul cu coudl 6
SELECT "employee_id", "first_name", "last_name"
FROM employees_ast e
WHERE e."employee_id" >= 6 AND NOT EXISTS (SELECT 1
                    FROM shops_ast s
                    WHERE "employee_id" = 6
                    AND NOT EXISTS (SELECT 1
                            FROM shops_ast
                            WHERE "employee_id" = e."employee_id" AND "shop_id" = s."shop_id"))
ORDER BY "employee_id";
```

Query Result

SQL | All Rows Fetched: 2 in 0.013 seconds

	employee_id	first_name	last_name
1	6	Miruna	Irinel
2	7	Ana	Constantin