Université de Sciences et Téchnologies Houari Boumedien

Faculty d'Informatique
Master 1 RSD

TP Reseaux BGP

TP Reseaux et Protocols

Prof: Mme Bouachi Farida

Etudiant: Belmouloud Mustapha Abdellah

Matricule: 212131092524

Groupe: 1

December 26, 2025

1 i hate this fucking subject, fucking whale of a teacher giving me assignmments during vacation FUCK YOU

2 TP1

1. create tablespace belmouloud_tbs datafile '/opt/oracle/oradata/XE/belmouloud_tbs.dbf' SIZE 100M AUTOEXTEND ON;

```
SQL> create tablespace belmouloud_tbs datafile '/opt/oracle/oradata/XE/belmouloud_tbs.dbf' SIZE 100M AUTOEXTEND ON;

Tablespace created.
```

   create temporary tablespace belmouloud2_temptbs tempfile '/opt/oracle/oradata/XE/belmouloud2_temptbs.dbf' SIZE 100M AUTOEXTEND ON;

```
SQL> create temporary tablespace belmouloud2_temptbs tempfile '/opt/oracle/oradata/XE/belmouloud2_temptbs.dbf' SIZE 100M AUTOEXTEND ON

Tablespace created.
```

2. create user tp1_belmouloud identified by musmus3108 default tablespace belmouloud_tbs temporary tablespace belmouloud2_temptbs;

```
SQL> create user tp1_belmouloud identified by musmus3108 default tablespace belmouloud_tbs temporary tablespace belmouloud2_temptbs;

User created.
```

3. GRANT ALL privileges to tp1_belmouloud;

```
SQL> GRANT ALL privileges to tp1_belmouloud;

Grant succeeded.
```

4. les tableaux proposés:

   user(id primary key, name)

   sellers(id primary key, name)

   products(id primary key, name, seller_id foreign key, price(entre 20 et 100), category (tshirt, hoodie, jacket))

   orders(product_id foreign key, user_id foreign key, primarykey(product_id, use_id))

   order_comments (id primary key, product_id foreign key, user_id foreign key,comments)

   seller_reviews (id primary_key, seller_id foeignkey,user_id foreignkey,rating)

5. CREATE TABLE users (id NUMBER, name VARCHAR2(50), CONSTRAINT pk_users PRIMARY KEY (id));

   CREATE TABLE sellers (id NUMBER, name VARCHAR2(50), CONSTRAINT pk_sellers PRIMARY KEY (id));

   CREATE TABLE products (id NUMBER, name VARCHAR2(20), seller_id NUMBER, price NUMBER, category VARCHAR2(10), CONSTRAINT pk_products PRIMARY KEY (id), CONSTRAINT chk_price CHECK (price BETWEEN 20 AND 100), CONSTRAINT chk_category CHECK (category IN ('tshirt', 'hoodie', 'jacket')), CONSTRAINT fk_product_seller FOREIGN KEY (seller_id) REFERENCES sellers (id));

   CREATE TABLE orders (product_id NUMBER, user_id NUMBER, CONSTRAINT pk_orders PRIMARY KEY (product_id, user_id), CONSTRAINT fk_order_product FOREIGN KEY (product_id) REFERENCES products (id), CONSTRAINT fk_order_user FOREIGN KEY (user_id) REFERENCES users (id));

   CREATE TABLE order_comments (id NUMBER, product_id NUMBER, user_id NUMBER, commenttxt VARCHAR2(100), CONSTRAINT pk_order_comments PRIMARY KEY (id), CONSTRAINT fk_comment_order1 FOREIGN KEY (product_id) REFERENCES products(id), CONSTRAINT fk_comment_order2 FOREIGN KEY (user_id) REFERENCES users(id));

   CREATE TABLE seller_reviews (id NUMBER, seller_id NUMBER, user_id NUMBER, rating NUMBER, CONSTRAINT pk_seller_reviews PRIMARY KEY (id), CONSTRAINT chk_rating CHECK (rating BETWEEN 1 AND 5), CONSTRAINT fk_review_seller FOREIGN KEY (seller_id) REFERENCES sellers (id), CONSTRAINT fk_review_user FOREIGN KEY (user_id) REFERENCES users (id));

```
SQL> CREATE TABLE users (
id NUMBER,
name VARCHAR2(50),
CONSTRAINT pk_users PRIMARY KEY (id)
);  2    3    4    5

Table created.

SQL> CREATE TABLE sellers (
id NUMBER,
name VARCHAR2(50),
CONSTRAINT pk_sellers PRIMARY KEY (id)
);  2    3    4    5

Table created.

SQL> CREATE TABLE products (
id NUMBER,
name VARCHAR2(20),
seller_id NUMBER,
price NUMBER,
category VARCHAR2(10),
CONSTRAINT pk_products PRIMARY KEY (id),
CONSTRAINT chk_price CHECK (price BETWEEN 20 AND 100),
CONSTRAINT chk_category CHECK (category IN ('tshirt', 'hoodie', 'jacket')),
CONSTRAINT  2   fk_product_seller FOREIGN KEY (seller_id) REFERENCES sellers (id)
);  3    4    5    6    7    8    9   10   11

Table created.
```

```
SQL> CREATE TABLE orders (
product_id NUMBER,
user_id    NUMBER,
CONSTRAINT pk_orders PRIMARY KEY (product_id, user_id),
CONSTRAINT fk_order_product FOREIGN KEY (product_id) REFERENCES products (id),
CONSTRAINT fk_order_user FOREIGN KEY (user_id) REFERENCES users (id)
);  2    3    4    5    6    7

Table created.
```

```
SQL> CREATE TABLE order_comments (
id         NUMBER,
product_id NUMBER,
user_id    NUMBER,
commenttxt    VARCHAR2(100),
CONSTRAINT pk_order_comments PRIMARY KEY (id),
CONSTRAINT fk_comment_order1 FOREIGN KEY (product_id) REFERENCES products(id),
CONSTRAINT fk_comment_order2 FOREIGN KEY (user_id) REFERENCES users(id)
);  2    3    4    5    6    7    8    9

Table created.

SQL> CREATE TABLE seller_reviews (
id        NUMBER,
seller_id NUMBER,
user_id   NUMBER,
rating    NUMBER,
CONSTRAINT pk_seller_reviews PRIMARY KEY (id),
CONSTRAINT chk_rating CHECK (rating BETWEEN 1 AND 5),
CONSTRAINT fk_review_seller FOREIGN KEY (seller_id) REFERENCES sellers (id),
CONSTRAINT fk_review_user FOREIGN KEY (user_id) REFERENCES users (id)
);  2    3    4    5    6    7    8    9   10

Table created.
```

6. alter table users add lastname varchar2(20);

```
SQL> alter table users add lastname varchar(20);

Table altered.

SQL> desc users;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ID                                        NOT NULL NUMBER
 NAME                                               VARCHAR2(50)
 LASTNAME                                           VARCHAR2(20)
```

7. alter table users modify name varchar2(50) not null;
   alter table sellers modify name varcha02r(50) not null;

```
SQL> desc sellers;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ID                                        NOT NULL NUMBER
 NAME                                               VARCHAR2(50)

SQL> desc users;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ID                                        NOT NULL NUMBER
 NAME                                               VARCHAR2(50)
 LASTNAME                                           VARCHAR2(20)

SQL> alter table users modify name varchar(50) not null;

Table altered.

SQL> alter table sellers modify name varchar(50) not null;

Table altered.
```

8. alter table users modify name varchar2(100);

```
SQL> alter table users modify name varchar(100);

Table altered.

SQL> desc users;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ID                                        NOT NULL NUMBER
 NAME                                      NOT NULL VARCHAR2(100)
 LASTNAME                                           VARCHAR2(20)
```

9. alter table users drop column lastname;

```
SQL> desc users;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ID                                        NOT NULL NUMBER
 NAME                                      NOT NULL VARCHAR2(100)
 LASTNAME                                           VARCHAR2(20)

SQL> alter table users drop column lastname;

Table altered.

SQL> desc users;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ID                                        NOT NULL NUMBER
 NAME                                      NOT NULL VARCHAR2(100)
```

10. alter table users rename column name to fullname;

```
SQL> desc users;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ID                                        NOT NULL NUMBER
 NAME                                      NOT NULL VARCHAR2(100)

SQL> alter table users rename column name to fullname;

Table altered.

SQL> desc users;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ID                                        NOT NULL NUMBER
 FULLNAME                                  NOT NULL VARCHAR2(100)
```

11. alter table orders add constraint fk_order_seller foreign key (seller_id) references sellers(id);

```
SQL> alter table orders add seller_id number;

Table altered.

SQL> alter table orders add constraint fk_order_seller foreign key (seller_id) references sellers(id);

Table altered.
```

12. alter table products add constraint dlp check(discount ¡ price);

```
SQL> alter table products add discount number;

Table altered.

SQL> alter table products add constraint dlp check(discount < price);

Table altered.
```

13. CREATE OR REPLACE TRIGGER potr BEFORE INSERT OR UPDATE ON orders FOR EACH ROW DECLARE vdiscount products.discountBEGIN SELECT discount INTO vdiscount FROM products WHERE id = :NEW.product_id; IF :NEW.discounted != vdiscount THEN RAISE_APPLICATION_ERROR(-20001, 'Order discount must match product discount'); END IF; END; /

```
SQL> CREATE OR REPLACE TRIGGER potr
BEFORE INSERT OR UPDATE ON orders
FOR EACH ROW
DECLARE
  vdiscount products.discount%TYPE;
BEGIN
  SELECT discount
  INTO vdiscount
  FROM products
  WHERE id = :NEW.product_id;
  IF :NEW.discounted != v  2    3  discount THEN
    RAISE_APPLICATION_ERROR(-20001, 'Order discount must match product discount');
  END IF;
END;
/  4    5    6    7    8    9   10   11   12   13   14   15

Trigger created.
```

14. INSERT INTO users (id, fullname) VALUES (1, 'Alice Johnson');
INSERT INTO users (id, fullname) VALUES (2, 'Bob Smith');
INSERT INTO users (id, fullname) VALUES (3, 'Charlie Brown');
INSERT INTO users (id, fullname) VALUES (4, 'Diana Prince');
INSERT INTO users (id, fullname) VALUES (5, 'Ethan Hunt');
INSERT INTO users (id, fullname) VALUES (6, 'Fiona Apple');
INSERT INTO users (id, fullname) VALUES (7, 'George Martin');

INSERT INTO sellers (id, name) VALUES (1, 'CoolTees');
INSERT INTO sellers (id, name) VALUES (2, 'HoodieHub');
INSERT INTO sellers (id, name) VALUES (3, 'JacketWorld');
INSERT INTO sellers (id, name) VALUES (4, 'UrbanWear');
INSERT INTO sellers (id, name) VALUES (5, 'FashionFiesta');
INSERT INTO sellers (id, name) VALUES (6, 'TrendSetters');
INSERT INTO sellers (id, name) VALUES (7, 'StreetStyle');

INSERT INTO products (id, name, seller_id, price, category, discount) VALUES (1, 'Classic Tee', 1, 25, 'tshirt', 5);
INSERT INTO products (id, name, seller_id, price, category, discount) VALUES (2, 'Premium Hoodie', 2, 60, 'hoodie', 10);
INSERT INTO products (id, name, seller_id, price, category, discount) VALUES (3, 'Leather Jacket', 3, 90, 'jacket', 15);
INSERT INTO products (id, name, seller_id, price, category, discount) VALUES (4, 'Graphic Tee', 1, 30, 'tshirt', 7);
INSERT INTO products (id, name, seller_id, price, category, discount) VALUES (5, 'Zip Hoodie', 2, 55, 'hoodie', 12);
INSERT INTO products (id, name, seller_id, price, category, discount) VALUES (6, 'Winter Jacket', 3, 95, 'jacket', 20);
INSERT INTO products (id, name, seller_id, price, category, discount) VALUES (7, 'Summer Tee', 1, 22, 'tshirt', 0);

INSERT INTO orders (product_id, user_id, seller_id, discounted) VALUES (1, 1, 1,

5);
INSERT INTO orders (product_id, user_id, seller_id, discounted) VALUES (2, 2, 2, 10);
INSERT INTO orders (product_id, user_id, seller_id, discounted) VALUES (3, 3, 3, 15);
INSERT INTO orders (product_id, user_id, seller_id, discounted) VALUES (4, 4, 1, 7);
INSERT INTO orders (product_id, user_id, seller_id, discounted) VALUES (5, 5, 2, 12);
INSERT INTO orders (product_id, user_id, seller_id, discounted) VALUES (6, 6, 3, 20);
INSERT INTO orders (product_id, user_id, seller_id, discounted) VALUES (7, 7, 1, 0);

INSERT INTO order_comments (id, product_id, user_id, commenttxt) VALUES (1, 1, 1, 'Great quality!');
INSERT INTO order_comments (id, product_id, user_id, commenttxt) VALUES (2, 2, 2, 'Very cozy hoodie.');
INSERT INTO order_comments (id, product_id, user_id, commenttxt) VALUES (3, 3, 3, 'Love the jacket!');
INSERT INTO order_comments (id, product_id, user_id, commenttxt) VALUES (4, 4, 4, 'Nice graphic design.');
INSERT INTO order_comments (id, product_id, user_id, commenttxt) VALUES (5, 5, 5, 'Good value for price.');
INSERT INTO order_comments (id, product_id, user_id, commenttxt) VALUES (6, 6, 6, 'Perfect for winter.');
INSERT INTO order_comments (id, product_id, user_id, commenttxt) VALUES (7, 7, 7, 'Light and comfortable.');

INSERT INTO seller_reviews (id, seller_id, user_id, rating) VALUES (1, 1, 1, 5);
INSERT INTO seller_reviews (id, seller_id, user_id, rating) VALUES (2, 2, 2, 4);
INSERT INTO seller_reviews (id, seller_id, user_id, rating) VALUES (3, 3, 3, 5);
INSERT INTO seller_reviews (id, seller_id, user_id, rating) VALUES (4, 1, 4, 3);
INSERT INTO seller_reviews (id, seller_id, user_id, rating) VALUES (5, 2, 5, 4);
INSERT INTO seller_reviews (id, seller_id, user_id, rating) VALUES (6, 3, 6, 5);
INSERT INTO seller_reviews (id, seller_id, user_id, rating) VALUES (7, 1, 7, 4);

la requete "insert into orders (product_id, user_id, seller_id, discounted) values (1, 2, 1, 999);" vas echouer a cause de trigger deja declaré qui necessite que discount = discounted

```
SQL> insert into orders (product_id, user_id, seller_id, discounted) values (1, 2, 1, 999);
insert into orders (product_id, user_id, seller_id, discounted) values (1, 2, 1, 999)
            *
ERROR at line 1:
ORA-20001: Order discount must match product discount
ORA-06512: at "SYSTEM.POTR", line 9
ORA-04088: error during execution of trigger 'SYSTEM.POTR'
```

la requete "INSERT INTO products (id, name, seller_id, price, category, discount) VALUES (8, 'Expensive Tee', 1, 150, 'tshirt', 10); " vas echouer a cause constarint check qui necessite price¿discount

```
SQL> INSERT INTO products (id, name, seller_id, price, category, discount)
VALUES (8, 'Expensive Tee', 1, 150, 'tshirt', 10);
  2  INSERT INTO products (id, name, seller_id, price, category, discount)
*
ERROR at line 1:
ORA-02290: check constraint (SYSTEM.CHK_PRICE) violated
```

INSERT INTO products (id, name, seller_id, price, category, discount) VALUES (9, 'Fancy Pants', 2, 50, 'pants', 5); vas echouer car la category doit etre tshirt hoodie ou jacket

```
SQL> INSERT INTO products (id, name, seller_id, price, category, discount)
VALUES (9, 'Fancy Pants', 2, 50, 'pants', 5);
  2  INSERT INTO products (id, name, seller_id, price, category, discount)
*
ERROR at line 1:
ORA-02290: check constraint (SYSTEM.CHK_CATEGORY) violated
```

15. alter table products disable constraint chk_price;

16. delete from products where id=1;
    contrainte d'integrité car on a pas specifié de qui fair on delete.

```
SQL> delete from products where id=1;
delete from products where id=1
*
ERROR at line 1:
ORA-02292: integrity constraint (SYSTEM.FK_COMMENT_ORDER1) violated - child
record found
```

17. assuré

18. select * from products, orders, users where orders.product_id = products.id and orders.user_id = users.id and products.discount ¿10;

```
SQL> select * from products, orders, users where orders.product_id = products.id and orders.user_id = users.id and products.discount >10;

        ID NAME                 SELLER_ID      PRICE CATEGORY     DISCOUNT
---------- -------------------- ---------- ---------- ---------- ----------
PRODUCT_ID    USER_ID  SELLER_ID DISCOUNTED         ID
---------- ---------- ---------- ---------- ----------
FULLNAME
--------------------------------------------------------------------------------
         3 Leather Jacket               3         90 jacket             15
         3          3          3         15          3
Charlie Brown

         5 Zip Hoodie                   2         55 hoodie             12
         5          5          2         12          5
Ethan Hunt

        ID NAME                 SELLER_ID      PRICE CATEGORY     DISCOUNT
---------- -------------------- ---------- ---------- ---------- ----------
PRODUCT_ID    USER_ID  SELLER_ID DISCOUNTED         ID
---------- ---------- ---------- ---------- ----------
FULLNAME
--------------------------------------------------------------------------------

         6 Winter Jacket                3         95 jacket             20
         6          6          3         20          6
Fiona Apple
```

19. select seller_id , avg(discounted) as average_discount from orders group by seller_id;

```
SQL> select seller_id , avg(discounted) as average_discount from orders group by seller_id;

 SELLER_ID AVERAGE_DISCOUNT
---------- ----------------
         1                4
         2               11
         3             17.5
```

20. create view qst20 as select seller_id , avg(discounted) as average_discount from orders group by seller_id;

```
SQL> create view qst20 as select seller_id , avg(discounted) as average_discount from orders group by seller_id;

View created.

SQL> select * from qst20;

 SELLER_ID AVERAGE_DISCOUNT
---------- ----------------
         1                4
         2               11
         3             17.5
```

3 Resultas

Example avec shiw ip route

Figure 1: Example Routeur OSPF