BOOKSTORE

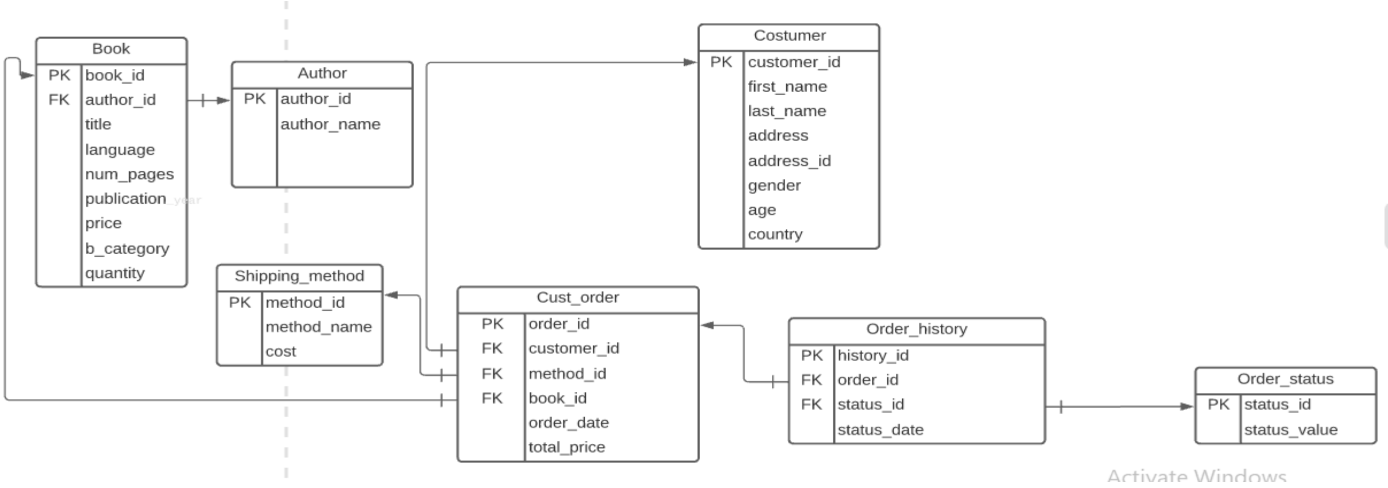
Prepared By:

- Ahmed Ben Abdallah
- Mohamed Mbarki

- **Purpose:**

During the latest global pandemic, the demand for books has risen exponentially with most people opting for soft copies ordered from online bookstores due to retail bookstores closing their doors .To manage these bookstores we must implement a sophisticated yet efficient management system. This management system will help in maintaining the details of the books available, the authors, the costumers and information about orders.

- **Conceptual Schema:**



- **Relational Schema:**

- book (**book_id**, #author_id, title, language, num_pages, publication_year, price, b_category, quantity)

- customer (**customer_id**, #address_id, first_name, last_name, address, , gender, age, country)

- shipping_method (**method_id**, method_name , cost)

- order_status (**status_id**, status_value)

- cust_order (**order_id** , #customer_id, #book_id , #method_id ,order_date , total_price)

- order_history (**history_id**, #order_id, #status_id , status_date)

- **Procedures:**

- nb_book_sold_per_quarter : Display the number of books sold for every quarter.

- revenue_per_quarter: Display the revenue for every quarter.

- nb_oders_per_status : Display the number of order for every order status.

- nb_books_buought_by_country : Display the number of books bought by every country.

- top_buyers: Display the list of top buyers.

- list_customer_orders: Display the list of customers with their order.

- nb_copy_sold_per_book: Display the number of copies sold for every book.

- nb_books_per_method: Display the number of books sold using every method.

- best_seller_book: Displat the best seller book.

- **Functions:**

printf_nb_book_per_gender : Display the number of books bought by gender. It takes gender name as a parameter.

nb_order_in_period: Display number of books in a given period of time. It takes two dates as a parameter.

total_spending_per_customer: Display the total Spending for every customer. It takes customer's id as a parameter.

- **Triggers:**

updatetables: Trigger to update order_history and book when inserting new order.

check_book_stock: Check book quantity before inserting new order . If the stock of the book is 0 it will raise an error message "ERROR : quantity = 0 , book out of stock".

update_deleted_order: update book and order_history tables when deleting the order , so it will add a book to the quantity column and delete the order from order history.

- **Packages:**

bookstore_package: contain all procedures and functions to manage the system.

- **Conclusion:**

The bookstore management system makes life easier for bookstore owners. It Targets problems in the bookstore and optimizes processes to achieve higher sales and customer satisfaction .

- **Login information:**

Username : homeuser

Password : password2

- **Code:**

```
/* DROP TABLES */
drop table "HOMEUSER"."AUTHOR" cascade constraints PURGE;
drop table "HOMEUSER"."BOOK" cascade constraints PURGE;
drop table "HOMEUSER"."CUSTOMER" cascade constraints PURGE;
drop table "HOMEUSER"."SHIPPING_METHOD" cascade constraints PURGE;
drop table "HOMEUSER"."CUST_ORDER" cascade constraints PURGE;
drop table "HOMEUSER"."ORDER_STATUS" cascade constraints PURGE;
drop table "HOMEUSER"."ORDER_HISTORY" cascade constraints PURGE;

/* TABLE AUTHTOR */
create table author
(
   author_id NUMBER(2),
   author_name VARCHAR2(100),
   CONSTRAINT pk_author_ahmed_ben PRIMARY KEY (author_id)
```

```
);

/*TABLE BOOK*/
CREATE TABLE book (
   book_id NUMBER(2),
   title VARCHAR2(400),
   language varchar2(30),
   num_pages NUMBER(5),
   publication_year NUMBER(4),
         price DECIMAL(6,2),
         author_id NUMBER(2),
         b_category varchar2(50),
         quantity NUMBER(5),
   CONSTRAINT pk_book PRIMARY KEY (book_id),
         CONSTRAINT fk_author FOREIGN KEY (author_id) REFERENCES author (author_id)
);

/*TABLE CUSTOMER*/
CREATE TABLE customer (
   customer_id NUMBER(2),
   first_name VARCHAR2(200),
   last_name VARCHAR2(200),
         address VARCHAR2(30),
         address_id NUMBER(2),
        gender VARCHAR2(20),
        age NUMBER(2),
        country VARCHAR2(30),
   CONSTRAINT pk_customer PRIMARY KEY (customer_id)
);

 /* TABLE SHIPPING METHOD*/
CREATE TABLE shipping_method (
   method_id NUMBER(2),
   method_name VARCHAR2(100),
   cost DECIMAL(10, 2),
   CONSTRAINT pk_shipmethod PRIMARY KEY (method_id)
);

/*TABLE ORDER_STATUS*/
 CREATE TABLE order_status (
   status_id NUMBER(2),
   status_value VARCHAR2(20),
   CONSTRAINT pk_orderstatus PRIMARY KEY (status_id)
);
```

```
/*TABLE CUSTOMER ORDER*/
CREATE TABLE cust_order (
   order_id NUMBER(2),
   order_date DATE,
   customer_id NUMBER,
         book_id NUMBER(2),
         method_id NUMBER(2),
         total_price decimal(6,2),
   CONSTRAINT pk_custorder PRIMARY KEY (order_id),
   CONSTRAINT fk_order_cust12 FOREIGN KEY (customer_id) REFERENCES customer (customer_id),
         CONSTRAINT fk_ol_book15 FOREIGN KEY (book_id) REFERENCES book (book_id),
         CONSTRAINT fk_ol_shipping FOREIGN KEY (method_id) REFERENCES shipping_method (method_id)
);
```

```
/*TABLE ORDER HISTORY*/
CREATE TABLE order_history (
   history_id NUMBER(2),
   order_id NUMBER(2),
   status_id NUMBER(2),
   status_date DATE,
   CONSTRAINT pk_orderhist PRIMARY KEY (history_id),
   CONSTRAINT fk_oh_order FOREIGN KEY (order_id) REFERENCES cust_order (order_id),
   CONSTRAINT fk_oh_status FOREIGN KEY (status_id) REFERENCES
order_status (status_id)
);
```

- **TABLE INSERTION:**

```
/** TABLE AUTHOR **/
insert into author values(1,'Charles Dickens');
insert into author values(2,'J. R. R. Tolkien');
insert into author values(3,'J. K. Rowling');
insert into author values(4,'Antoine de Saint-Exupéry');
insert into author values(5,'Cao Xueqin');
insert into author values(6,'Agatha Christie');
insert into author values(7,'C. S. Lewis');
insert into author values(8,'H. Rider Haggard');
insert into author values(9,'Carlo Collodi');
insert into author values(10,'  Dan Brown');
insert into author values(11,'  Paulo Coelho');
insert into author values(12,'J. D. Salinger');
insert into author values(13,'Robert James Waller');
insert into author values(14,'Lew Wallace');
insert into author values(15,'Louise Hay');
insert into author values(16,'Gabriel García Márquez');
insert into author values(17,'Vladimir Nabokov');
insert into author values(18,'Johanna Spyri');
insert into author values(19,'Benjamin Spock');
insert into author values(20,'Lucy Maud Montgomery');

insert into author values(21,'Anna Sewell');

insert into author values(22,'Fyodor Dostoevsky');

insert into author values(23,'George Orwell');


/** TABLE BOOK **/


Insert into BOOK
(BOOK_ID,TITLE,LANGUAGE,NUM_PAGES,PUBLICATION_YEAR,PRICE,AUTHO
R_ID,B_CATEGORY,QUANTITY) values ('1','A Tale of Two
Cities','English','448','1859','10','1','Historical Literature','2');

Insert into BOOK
(BOOK_ID,TITLE,LANGUAGE,NUM_PAGES,PUBLICATION_YEAR,PRICE,AUTHO
R_ID,B_CATEGORY,QUANTITY) values ('2','The
Hobbit','English','310','1937','8','2','Fantasy','7');

Insert into BOOK
(BOOK_ID,TITLE,LANGUAGE,NUM_PAGES,PUBLICATION_YEAR,PRICE,AUTHO
R_ID,B_CATEGORY,QUANTITY) values ('3',' Harry Potter and the
Philosopher''s Stone','English','384','1997','11','3','Fantasy','3');

Insert into BOOK
(BOOK_ID,TITLE,LANGUAGE,NUM_PAGES,PUBLICATION_YEAR,PRICE,AUTHO
R_ID,B_CATEGORY,QUANTITY) values ('4',' The Little Prince
        ','English','442','1943','13','4','Novella','0');

Insert into BOOK
(BOOK_ID,TITLE,LANGUAGE,NUM_PAGES,PUBLICATION_YEAR,PRICE,AUTHO
R_ID,B_CATEGORY,QUANTITY) values ('5',' Dream of the Red Chamber
        ','Chinese','698','1762','16','5','Family saga','6');

Insert into BOOK
(BOOK_ID,TITLE,LANGUAGE,NUM_PAGES,PUBLICATION_YEAR,PRICE,AUTHO
R_ID,B_CATEGORY,QUANTITY) values ('6','And Then There Were
None','English','386','1939','9','6','Mystery','4');
```

```
Insert into BOOK
(BOOK_ID,TITLE,LANGUAGE,NUM_PAGES,PUBLICATION_YEAR,PRICE,AUTHOR_ID,B_CATEGORY,QUANTITY) values ('7','The Lion, the Witch and the Wardrobe    ','English','561','1950','14','7','Fantasy','2');

Insert into BOOK
(BOOK_ID,TITLE,LANGUAGE,NUM_PAGES,PUBLICATION_YEAR,PRICE,AUTHOR_ID,B_CATEGORY,QUANTITY) values ('8','She: A History of Adventure        ','English','429','1887','21','8','Adventure','2');

Insert into BOOK
(BOOK_ID,TITLE,LANGUAGE,NUM_PAGES,PUBLICATION_YEAR,PRICE,AUTHOR_ID,B_CATEGORY,QUANTITY) values ('9','The Adventures of Pinocchio        ','Italian','121','1881','8','9','Fantasy','6');

Insert into BOOK
(BOOK_ID,TITLE,LANGUAGE,NUM_PAGES,PUBLICATION_YEAR,PRICE,AUTHOR_ID,B_CATEGORY,QUANTITY) values ('10','The Da Vinci Code        ','English','725','2003','16','10','Mystery thriller','4');

Insert into BOOK
(BOOK_ID,TITLE,LANGUAGE,NUM_PAGES,PUBLICATION_YEAR,PRICE,AUTHOR_ID,B_CATEGORY,QUANTITY) values ('11',' Harry Potter and the Chamber of Secrets','English','313','1998','11,5','3','Fantasy','3');

Insert into BOOK
(BOOK_ID,TITLE,LANGUAGE,NUM_PAGES,PUBLICATION_YEAR,PRICE,AUTHOR_ID,B_CATEGORY,QUANTITY) values ('12',' Harry Potter and the Prisoner of Azkaban','English','438','1999','12','3','Fantasy','6');

Insert into BOOK
(BOOK_ID,TITLE,LANGUAGE,NUM_PAGES,PUBLICATION_YEAR,PRICE,AUTHOR_ID,B_CATEGORY,QUANTITY) values ('13',' Harry Potter and the Goblet of Fire','English','287','2000','10','3','Fantasy','5');

Insert into BOOK
(BOOK_ID,TITLE,LANGUAGE,NUM_PAGES,PUBLICATION_YEAR,PRICE,AUTHOR_ID,B_CATEGORY,QUANTITY) values ('14','Ben-Hur: A Tale of the Christ','English','516','1880','7,7','14','Novel','9');
```

```
Insert into BOOK (BOOK_ID,TITLE,LANGUAGE,NUM_PAGES,PUBLICATION_YEAR,PRICE,AUTHOR_ID,B_CATEGORY,QUANTITY) values ('15','You Can Heal Your Life','English','272','1984','22,49','15','Self-help','7');

Insert into BOOK (BOOK_ID,TITLE,LANGUAGE,NUM_PAGES,PUBLICATION_YEAR,PRICE,AUTHOR_ID,B_CATEGORY,QUANTITY) values ('16','One Hundred Years of Solitude','Spanish','448','1967','57,6','16','Magic realism','8');

Insert into BOOK (BOOK_ID,TITLE,LANGUAGE,NUM_PAGES,PUBLICATION_YEAR,PRICE,AUTHOR_ID,B_CATEGORY,QUANTITY) values ('17','Lolita','English','336','1955','16,65','17','Novel','2');

Insert into BOOK (BOOK_ID,TITLE,LANGUAGE,NUM_PAGES,PUBLICATION_YEAR,PRICE,AUTHOR_ID,B_CATEGORY,QUANTITY) values ('18','Heidi','German','148','1880','5,79','18','Children fiction','4');

Insert into BOOK (BOOK_ID,TITLE,LANGUAGE,NUM_PAGES,PUBLICATION_YEAR,PRICE,AUTHOR_ID,B_CATEGORY,QUANTITY) values ('19','The Common Sense Book of Baby and Child Care','English','208','1946','12,5','19','Manual','4');

Insert into BOOK (BOOK_ID,TITLE,LANGUAGE,NUM_PAGES,PUBLICATION_YEAR,PRICE,AUTHOR_ID,B_CATEGORY,QUANTITY) values ('20','Anne of Green Gables','English','208','1908','7,79','20','Children novel','1');

Insert into BOOK (BOOK_ID,TITLE,LANGUAGE,NUM_PAGES,PUBLICATION_YEAR,PRICE,AUTHOR_ID,B_CATEGORY,QUANTITY) values ('21','Black Beauty','English','336','1877','7,67','21','Children literature','1');

Insert into BOOK (BOOK_ID,TITLE,LANGUAGE,NUM_PAGES,PUBLICATION_YEAR,PRICE,AUTHOR_ID,B_CATEGORY,QUANTITY) values ('22','Crime and Punishmen','Russian','700','1866','31,2','22','Novel','2');
Insert into BOOK (BOOK_ID,TITLE,LANGUAGE,NUM_PAGES,PUBLICATION_YEAR,PRICE,AUTHOR_ID,B_CATEGORY,QUANTITY) values ('23','Animal Farm','English','112','1945','9,99','23','Politics','6');


/** TABLE CUSTOMER **/

INSERT INTO customer VALUES (1,'Salah','Ben Amor','90 Bulgaria Sousse',1,'male',34,'Tunisia');

INSERT INTO customer VALUES (2,'Phill','Foden','452 St-pierre Paris',2,'male',21,'France');

INSERT INTO customer VALUES (3,'Bernardo','Silva','1024 liberation ave Kinshasa',3,'male',29,'Congo DCR');

INSERT INTO customer VALUES (4,'Luka','Jovic','1223 Sheikh salman st Kuwait',4,'female',22,'Kuwait');

INSERT INTO customer VALUES (5,'Mohamed','Issaoui','12354 madrid Tunis',5,'female',57,'Tunisia');

INSERT INTO customer VALUES (6,'Robert','Dinero','121154 charliton st Bermingham',6,'male',74,'United Kingdom');

INSERT INTO customer VALUES (7,'Sarah','Hakimi','65536 mohamed 6 marrakech',7,'female',18,'Morocco');

INSERT INTO customer VALUES (8,'Sadio','Mane','245451 victoria ave dakkar',8,'male',24,'Senegal');

INSERT INTO customer VALUES (9,'Neck','James','4026 ben franklin ave portland',9,'female',36,'United States');

INSERT INTO customer VALUES (10,'Lebron','Meth','7512 Goethenberg st Malmo,10',10,'female',42,'Sweden');

INSERT INTO customer VALUES (11,'Johan','Cruyff','Zwaansweg 15 Kedichem,Zuid-Holland',11,'male',60,'Netherlands');

INSERT INTO customer VALUES (12,'Martin','Odeegard','Askeveien 166,Oslo',12,'female',23,'Norway');
INSERT INTO customer VALUES (13,'Paul','Allen','4512 Liberty st New York',13,'male',26,'United States');

INSERT INTO customer VALUES (14,'Akram','Afif','8751 1973 ave Doha',14,'male',29,'Qatar');

INSERT INTO customer VALUES (15,'Selina','Sharapova','2154 Rasputin st Moscow',15,'female',34,'Russia');


/**TABLE SHIPPING METHOD**/

insert into shipping_method values (1,'UPS',89.99);

insert into shipping_method values (2,'FEDEX',253.82);

insert into shipping_method values (3,'USPS',230);

insert into shipping_method values (4,'INTERNATIONAL',34.99);


/**TABLE ODER_STATUS**/

insert into order_status values(1,'Pending');

insert into order_status values(2,'In Progress');

insert into order_status values(3,'Delivered');

insert into order_status values(4,'Cancelled');


/** TABLE CUSTOMER ORDER **/

Insert into CUST_ORDER (ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRICE) values ('1',to_date('05/01/21','DD/MM/RR'),'1','1','1','899,9');

Insert into CUST_ORDER (ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRICE) values ('2',to_date('10/01/21','DD/MM/RR'),'1','4','2','266,82');
Insert into CUST_ORDER (ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRICE) values ('3',to_date('22/02/21','DD/MM/RR'),'2','5','3','246');

Insert into CUST_ORDER (ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRICE) values ('4',to_date('20/03/21','DD/MM/RR'),'3','2','4','42,99');

Insert into CUST_ORDER (ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRICE) values ('5',to_date('12/03/21','DD/MM/RR'),'4','6','3','239');

Insert into CUST_ORDER (ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRICE) values ('6',to_date('17/04/21','DD/MM/RR'),'5','7','1','103,99');

Insert into CUST_ORDER (ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRICE) values ('7',to_date('07/05/21','DD/MM/RR'),'7','9','1','103,99');

Insert into CUST_ORDER (ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRICE) values ('8',to_date('21/05/21','DD/MM/RR'),'8','10','1','97,99');

Insert into CUST_ORDER (ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRICE) values ('9',to_date('25/05/21','DD/MM/RR'),'9','10','1','97,99');

Insert into CUST_ORDER (ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRICE) values ('10',to_date('10/06/21','DD/MM/RR'),'9','11','1','110,49');

Insert into CUST_ORDER (ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRICE) values ('11',to_date('01/06/21','DD/MM/RR'),'10','12','2','265,82');

Insert into CUST_ORDER (ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRICE) values ('12',to_date('12/07/21','DD/MM/RR'),'10','13','3','240');
Insert into CUST_ORDER
(ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRIC
E) values ('13',to_date('16/07/21','DD/MM/RR'),'12','1','3','240');

Insert into CUST_ORDER
(ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRIC
E) values ('14',to_date('11/08/21','DD/MM/RR'),'10','1','3','240');

Insert into CUST_ORDER
(ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRIC
E) values ('15',to_date('25/08/21','DD/MM/RR'),'9','6','2','262,82');

Insert into CUST_ORDER
(ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRIC
E) values ('16',to_date('12/12/21','DD/MM/RR'),'3','6','2','262,82');

Insert into CUST_ORDER
(ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRIC
E) values ('17',to_date('20/10/21','DD/MM/RR'),'3','6','2','262,82');

Insert into CUST_ORDER
(ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRIC
E) values ('18',to_date('28/10/21','DD/MM/RR'),'3','2','3','238');

Insert into CUST_ORDER
(ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRIC
E) values ('19',to_date('17/11/21','DD/MM/RR'),'3','8','1','110,99');

Insert into CUST_ORDER
(ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRIC
E) values ('20',to_date('23/11/21','DD/MM/RR'),'3','12','1','265,82');

Insert into CUST_ORDER
(ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRIC
E) values ('21',to_date('30/11/21','DD/MM/RR'),'3','22','1','285,02');

Insert into CUST_ORDER
(ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRIC
E) values ('22',to_date('18/10/21','DD/MM/RR'),'4','22','1','285,02');
```

```
Insert into CUST_ORDER (ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRICE) values ('23',to_date('20/10/21','DD/MM/RR'),'2','22','1','285,02');

Insert into CUST_ORDER (ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRICE) values ('24',to_date('11/10/21','DD/MM/RR'),'4','22','3','261,2');

Insert into CUST_ORDER (ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRICE) values ('25',to_date('11/09/21','DD/MM/RR'),'4','22','3','261,2');

Insert into CUST_ORDER (ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRICE) values ('26',to_date('20/08/21','DD/MM/RR'),'4','23','3','239,99');

Insert into CUST_ORDER (ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRICE) values ('27',to_date('20/08/21','DD/MM/RR'),'12','21','4','42,66');

Insert into CUST_ORDER (ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRICE) values ('28',to_date('20/08/21','DD/MM/RR'),'5','21','4','42,66');

Insert into CUST_ORDER (ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRICE) values ('29',to_date('20/06/21','DD/MM/RR'),'5','20','2','261,61');

Insert into CUST_ORDER (ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRICE) values ('30',to_date('11/05/21','DD/MM/RR'),'6','20','2','261,61');

Insert into CUST_ORDER (ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRICE) values ('31',to_date('20/05/21','DD/MM/RR'),'11','19','3','242,5');

Insert into CUST_ORDER (ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRICE) values ('32',to_date('15/04/21','DD/MM/RR'),'9','19','3','242,5');
Insert into CUST_ORDER (ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRICE) values ('33',to_date('14/01/21','DD/MM/RR'),'9','13','1','99,99');

Insert into CUST_ORDER (ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRICE) values ('34',to_date('16/01/21','DD/MM/RR'),'12','13','1','99,99');

Insert into CUST_ORDER (ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRICE) values ('35',to_date('12/02/21','DD/MM/RR'),'8','11','2','265,32');

Insert into CUST_ORDER (ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRICE) values ('36',to_date('12/02/21','DD/MM/RR'),'11','11','2','265,32');

Insert into CUST_ORDER (ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRICE) values ('37',to_date('30/03/21','DD/MM/RR'),'4','6','4','43,99');

Insert into CUST_ORDER (ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRICE) values ('38',to_date('30/04/21','DD/MM/RR'),'5','22','4','66,19');

Insert into CUST_ORDER (ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRICE) values ('39',to_date('15/04/21','DD/MM/RR'),'5','4','2','266,82');

Insert into CUST_ORDER (ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRICE) values ('40',to_date('15/04/21','DD/MM/RR'),'10','4','2','266,82');

Insert into CUST_ORDER (ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRICE) values ('41',to_date('15/05/21','DD/MM/RR'),'3','4','2','266,82');

Insert into CUST_ORDER (ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRICE) values ('42',to_date('16/05/21','DD/MM/RR'),'3','8','3','251');
Insert into CUST_ORDER
(ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRIC
E) values ('43',to_date('16/06/21','DD/MM/RR'),'4','8','3','251');

Insert into CUST_ORDER
(ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRIC
E) values ('44',to_date('11/07/21','DD/MM/RR'),'3','22','3','261,2');

Insert into CUST_ORDER
(ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRIC
E) values ('45',to_date('16/07/21','DD/MM/RR'),'9','22','3','261,2');

Insert into CUST_ORDER
(ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRIC
E) values ('46',to_date('13/08/21','DD/MM/RR'),'9','6','3','239');

Insert into CUST_ORDER
(ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRIC
E) values ('47',to_date('22/08/21','DD/MM/RR'),'4','6','3','239');

Insert into CUST_ORDER
(ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRIC
E) values ('48',to_date('22/09/21','DD/MM/RR'),'1','6','3','239');

Insert into CUST_ORDER
(ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRIC
E) values ('49',to_date('27/10/21','DD/MM/RR'),'5','11','2','265,32');

Insert into CUST_ORDER
(ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRIC
E) values ('50',to_date('10/10/21','DD/MM/RR'),'10','11','2','265,32');

Insert into CUST_ORDER
(ORDER_ID,ORDER_DATE,CUSTOMER_ID,BOOK_ID,METHOD_ID,TOTAL_PRIC
E) values ('51',to_date('23/07/21','DD/MM/RR'),'4','2','2','261,82');


/** TABLE ORDER_HISTORY **/
```

```
Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('1','1','3',to_date('06/01/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('2','2','1',to_date('10/01/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('3','2','1',to_date('22/02/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('4','4','3',to_date('27/03/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('5','5','1',to_date('12/03/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('6','6','3',to_date('22/04/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('7','7','3',to_date('12/05/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('8','8','1',to_date('21/05/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('9','9','2',to_date('26/05/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('10','10','1',to_date('10/06/21','DD/MM/RR'));
```

```
Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('11','11','3',to_date('02/06/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('12','12','3',to_date('14/07/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('13','13','2',to_date('17/07/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('14','14','2',to_date('12/08/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('15','15','1',to_date('25/08/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('16','16','2',to_date('14/12/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('17','17','3',to_date('21/10/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('18','18','3',to_date('30/10/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('19','19','1',to_date('17/11/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('20','20','1',to_date('23/11/21','DD/MM/RR'));
```

```
Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('21','21','1',to_date('30/11/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('22','22','3',to_date('23/10/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('23','23','3',to_date('21/10/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('24','24','3',to_date('13/10/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('25','25','1',to_date('11/09/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('26','26','3',to_date('24/08/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('27','27','3',to_date('24/08/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('28','28','3',to_date('24/08/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('29','29','1',to_date('20/06/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('30','30','1',to_date('11/05/21','DD/MM/RR'));
```

```
Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('31','31','3',to_date('22/05/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('32','32','1',to_date('15/04/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('33','33','1',to_date('14/01/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('34','34','2',to_date('17/01/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('35','35','3',to_date('13/02/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('36','36','3',to_date('13/02/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('37','37','1',to_date('30/03/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('38','38','3',to_date('06/05/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('39','39','1',to_date('15/04/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('40','40','1',to_date('15/04/21','DD/MM/RR'));
```

```
Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('41','41','1',to_date('15/05/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('42','42','3',to_date('18/05/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('43','43','3',to_date('18/06/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('44','44','1',to_date('11/07/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('45','45','2',to_date('17/07/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('46','46','1',to_date('13/08/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('47','47','3',to_date('24/08/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('48','48','1',to_date('22/09/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('49','49','3',to_date('28/10/21','DD/MM/RR'));

Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('50','50','1',to_date('10/10/21','DD/MM/RR'));
```

```
Insert into ORDER_HISTORY
(HISTORY_ID,ORDER_ID,STATUS_ID,STATUS_DATE) values
('51','4','1',to_date('23/07/21','DD/MM/RR'));
```

- **Procedures , Functions , Triggers , Packages:**

```
create or replace procedure list_customer_orders

as

cursor c1 is select cust_order.order_id , first_name , last_name from customer ,
cust_order where customer.customer_id = cust_order.customer_id;

rec_c c1%rowtype;

begin

open c1;

loop

fetch c1 into rec_c;

exit when c1%notfound;

DBMS_OUTPUT.PUT_LINE('Order Id: '||rec_c.order_id|| ', First name:
'||rec_c.first_name||', Last name: '||rec_c.last_name);

end loop;

EXCEPTION

WHEN OTHERS THEN

dbms_output.put_line(sqlcode);

dbms_output.put_line(sqlerrm);

end;

execute list_customer_orders;


/********************************** /
```

```
/* nb books bought by every gender*/

create or replace function printf_nb_book_per_gender(u_gender customer.gender%type)

return VARCHAR2

as

nb_book number(3);

n_result varchar(1000);

begin

DBMS_OUTPUT.PUT_LINE('********************NUMBER OF BOOKS BOUGHT PER GENDER***************** ');

select COUNT(book.title) into nb_book from book , cust_order , customer

where book.book_id = cust_order.book_id AND customer.customer_id = cust_order.customer_id and customer.gender = u_gender;

n_result := 'Number of books bought by '||u_gender||' = '||nb_book;

return n_result;

end printf_nb_book_per_gender;


accept m_gender prompt 'enter gender';

begin

dbms_output.put_line(printf_nb_book_per_gender(&m_gender));

end;


/********************************************************* /

/*NUMBER OF BOOKS SOLD PER QUARTER*/

CREATE OR REPLACE procedure nb_book_sold_per_quarter

as
```

```
cursor c1 is select CUST_ORDER.order_date as od , count(book.book_id) as nb_book from CUST_ORDER , book , order_history where
CUST_ORDER.book_id = book.book_id and order_history.status_id = 3 group by CUST_ORDER.order_date ;
rec_c c1%rowtype;
first_q NUMBER(2);
second_q NUMBER(2);
fourth_q NUMBER(2);
third_q NUMBER(2);
begin
first_q := 0;
second_q :=0;
third_q :=0;
fourth_q := 0;
open c1;
loop
fetch c1 into rec_c;
exit when c1%notfound;
IF rec_c.od BETWEEN to_date('01/01/2021','mm/dd/yyyy') AND to_date('03/31/2021','mm/dd/yyyy') THEN
first_q := first_q + 1;
elsif rec_c.od BETWEEN to_date('04/01/2021','mm/dd/yyyy') AND to_date('06/30/2021','mm/dd/yyyy') THEN
second_q := second_q +1;
elsif rec_c.od BETWEEN to_date('07/01/2021','mm/dd/yyyy') AND to_date('09/30/2021','mm/dd/yyyy') THEN
third_q := third_q +1;
```

```
elsif rec_c.od BETWEEN to_date('10/01/2021','mm/dd/yyyy') AND to_date('12/31/2021','mm/dd/yyyy') THEN

fourth_q := fourth_q +1;

END IF;

end loop;

close c1;

dbms_output.put_line('NUMBER OF BOOKS BOUGHT IN THE FIRST QUARTER');

dbms_output.put_line(first_q);

dbms_output.put_line('NUMBER OF BOOKS BOUGHT IN THE SECOND QUARTER');

dbms_output.put_line(second_q);

dbms_output.put_line('NUMBER OF BOOKS BOUGHT IN THE THIRD QUARTER');

dbms_output.put_line(third_q);

dbms_output.put_line('NUMBER OF BOOKS BOUGHT IN THE FOURTH QUARTER');

dbms_output.put_line(fourth_q);

EXCEPTION

WHEN OTHERS THEN

dbms_output.put_line(sqlcode);

dbms_output.put_line(sqlerrm);

end nb_book_sold_per_quarter;


execute nb_book_sold_per_quarter;

/***************************************************** /
```

```
/*REVENUES OF EVERY QUARTER - total sum of delivered books*/
CREATE OR REPLACE procedure revenue_per_quarter
as
cursor c1 is select CUST_ORDER.order_date as od , cust_order.total_price as
total_sum from CUST_ORDER , order_history where
cust_order.order_id = order_history.order_id and order_history.status_id = 3;
rec_c c1%rowtype;
first_q_revenue DECIMAL(10, 2);
second_q_revenue DECIMAL(10, 2);
fourth_q_revenue DECIMAL(10, 2);
third_q_revenue DECIMAL(10, 2);
begin
first_q_revenue := 0;
second_q_revenue :=0;
third_q_revenue :=0;
fourth_q_revenue := 0;
open c1;
loop
fetch c1 into rec_c;
exit when c1%notfound;
IF rec_c.od BETWEEN to_date('01/01/2021','dd/mm/yyyy') AND
to_date('31/03/2021','dd/mm/yyyy') THEN
first_q_revenue := first_q_revenue + rec_c.total_sum;
elsif rec_c.od BETWEEN to_date('01/04/2021','dd/mm/yyyy') AND
to_date('30/06/2021','dd/mm/yyyy') THEN
second_q_revenue := second_q_revenue + rec_c.total_sum;
```

```
elsif rec_c.od BETWEEN to_date('01/07/2021','dd/mm/yyyy') AND
to_date('30/09/2021','dd/mm/yyyy') THEN

third_q_revenue := third_q_revenue + rec_c.total_sum;

elsif rec_c.od BETWEEN to_date('01/10/2021','dd/mm/yyyy') AND
to_date('31/12/2021','dd/mm/yyyy') THEN

fourth_q_revenue := fourth_q_revenue + rec_c.total_sum;

END IF;

end loop;

close c1;

dbms_output.put_line('REVENUES OF THE FIRST QUARTER');

dbms_output.put_line(first_q_revenue||'$');

dbms_output.put_line('REVENUES OF THE SECOND QUARTER');

dbms_output.put_line(second_q_revenue||'$');

dbms_output.put_line('REVENUES OF THE THIRD QUARTER');

dbms_output.put_line(third_q_revenue||'$');

dbms_output.put_line('REVENUES OF THE FOURTH QUARTER');

dbms_output.put_line(fourth_q_revenue||'$');

EXCEPTION

WHEN OTHERS THEN

dbms_output.put_line(sqlcode);

dbms_output.put_line(sqlerrm);

end revenue_per_quarter;


execute revenue_per_quarter;


/****************************************/
```

```
/* nb of orders per status*/
create or replace procedure nb_oders_per_status
as
nb_orders NUMBER(3);
statusValue order_status.status_value%type;
cursor c1 is select order_status.status_value ,count(order_history.status_id) from order_history , order_status
where order_history.status_id = order_status.status_id group by order_status.status_value;

begin
open c1;
loop
fetch c1 into statusValue , nb_orders;
exit when c1%notfound;
DBMS_OUTPUT.PUT_LINE('Number of book of status [ '||statusValue||' ] = '||nb_orders);
end loop;
close c1;
EXCEPTION
WHEN OTHERS THEN
dbms_output.put_line(sqlcode);
dbms_output.put_line(sqlerrm);
end nb_oders_per_status;

execute nb_oders_per_status;
```

```
/*********************************************/

/*NUMBER OF BOOKS BOUGHT BY COUNTRY*/

CREATE OR REPLACE PROCEDURE nb_books_buought_by_country

as

cursor c1 is select customer.country as country , count(cust_order.order_id)
as nb_book from cust_order , customer

where customer.customer_id = cust_order.customer_id group by
customer.country ORDER BY nb_book DESC;

rec_c c1%rowtype;

begin

DBMS_OUTPUT.PUT_LINE('*************NUMBER OF BOOKS BOUGHT BY
EVERY COUNTRY*****************');

open c1;

loop

fetch c1 into rec_c;

exit when c1%notfound;

DBMS_OUTPUT.PUT_LINE('COUNTRY NAME : '||rec_c.country||' => NUMBER
OF BOOKS BOUGHT : '||rec_c.nb_book);

end loop;

close c1;

EXCEPTION

WHEN OTHERS THEN

dbms_output.put_line(sqlcode);

dbms_output.put_line(sqlerrm);

end nb_books_buought_by_country;
```

```
execute nb_books_buought_by_country;


/******************************************/

/* NB of orders between two dates*/

/*date format dd/mm/yyyy*/

CREATE OR REPLACE FUNCTION nb_order_in_period(d_start DATE , d_end DATE)

return VARCHAR2

AS

nb NUMBER(2);

result varchar(1200);

begin

select count(cust_order.order_id) into nb from cust_order where cust_order.order_date BETWEEN d_start and d_end;

result := 'NUMBER OF ORDER BETWEEN '||d_start||' AND '||d_end||' : '||nb;

return result;

end nb_order_in_period;


accept d_starts prompt 'date 1';

accept d_ends prompt 'date 2';

begin

DBMS_OUTPUT.PUT_LINE(nb_order_in_period(&d_starts,&d_ends));

end;


/********************************************/
```

```
/* TOTAL SPENDING FOR EVERY CUSTOMER*/

CREATE OR REPLACE FUNCTION total_spending_per_customer(c_id cust_order.customer_id%type)

return VARCHAR2

AS

result varchar2(1200);

begin

DBMS_OUTPUT.PUT_LINE('***********TOTAL SPENDING PER CUSTOMER**************');

for cur_var in (

select first_name , last_name , sum(total_price) as total_spending from cust_order , customer

where cust_order.customer_id = c_id and cust_order.customer_id = customer.customer_id group by first_name,last_name)

loop

result := cur_var.first_name||' '||cur_var.last_name||' has spent: '||cur_var.total_spending||'$';

end loop;

return result;

end total_spending_per_customer;


accept customer prompt 'Enter Customer ID';

begin

dbms_output.put_line(total_spending_per_customer(&customer));

end;


/******************************************/
```

```
/* Number of books bought by every customer*/

CREATE OR REPLACE PROCEDURE top_buyers

IS

begin

DBMS_OUTPUT.PUT_LINE('LIST OF TOP BUYERS');

for cur_var in (select customer.first_name as fn , count(cust_order.order_id) as nb_order from customer , cust_order where customer.customer_id = cust_order.customer_id

group by customer.first_name ORDER BY count(cust_order.order_id) DESC)

loop

DBMS_OUTPUT.PUT_LINE(cur_var.fn || '      '||cur_var.nb_order);

end loop;

EXCEPTION

WHEN OTHERS THEN

dbms_output.put_line(sqlcode);

dbms_output.put_line(sqlerrm);

end;


execute top_buyers;


/************************************/

/* NB copies sold per book*/

CREATE OR REPLACE PROCEDURE nb_copy_sold_per_book

AS

begin
```

```
for c1 in (select book.title as title , count(cust_order.order_id) as nb from cust_order , book
where cust_order.book_id = book.book_id group by book.title)
loop
dbms_output.put_line('Book Title: '||c1.title||' =>      Number of Copies: '||c1.nb);
end loop;
EXCEPTION
WHEN OTHERS THEN
dbms_output.put_line(sqlcode);
dbms_output.put_line(sqlerrm);
end nb_copy_sold_per_book;


execute nb_copy_sold_per_book;


/***************************************/
/* Number of book per shipping method*/
CREATE OR REPLACE PROCEDURE nb_books_per_method
AS
cursor c1 is select shipping_method.method_name as shipping_name ,count(cust_order.order_id) as nb_order from cust_order , shipping_method
where cust_order.method_id = shipping_method.method_id group by shipping_method.method_name;
rec_c c1%rowtype;
begin
open c1;
```

```
loop
fetch c1 into rec_c;
exit when c1%notfound;
DBMS_OUTPUT.PUT_LINE(rec_c.nb_order||' Order has been shipping with '||rec_c.shipping_name);
end loop;
close c1;
EXCEPTION
WHEN OTHERS THEN
dbms_output.put_line(sqlcode);
dbms_output.put_line(sqlerrm);
end nb_books_per_method;


execute nb_books_per_method;


/*****************************************/
/* best seller book*/
CREATE OR REPLACE PROCEDURE best_seller_book
IS
cursor c1 is select book.title, author.author_name , count(cust_order.book_id) as nb_of_books from cust_order , book,author
where cust_order.book_id = book.book_id AND author.author_id = book.author_id
group by book.title,author.author_name ORDER BY nb_of_books DESC;
c_rec c1%rowtype;
begin
```

```
open c1;
loop
fetch c1 into c_rec;
exit when c1%rowcount=2;
dbms_output.put_line('*****************BEST SELLER
BOOK*******************');
dbms_output.put_line('Title : '||c_rec.title);
dbms_output.put_line('Number of sales: '||c_rec.nb_of_books);
dbms_output.put_line('Author: '||c_rec.author_name);
end loop;
close c1;
EXCEPTION
WHEN OTHERS THEN
dbms_output.put_line(sqlcode);
dbms_output.put_line(sqlerrm);
end;

execute best_seller_book;

/*********************************************************/
/* Trigger to update order_history and book when inserting new order*/
create or replace trigger updatetables
after insert on cust_order
for each row
BEGIN
```

```
update book set quantity = quantity -1 where book_id =:new.book_id;
insert into order_history values
(:new.order_id,:new.customer_id,1,:new.order_date);
end if;
end;

/*Check book quantity*/
create or replace trigger check_book_stock
before insert on cust_order
for each row
DECLARE
qt NUMBER(2);
BEGIN
select quantity into qt from book where book_id = :new.book_id;
if qt = 0 then
   raise_application_error(-20030,'ERROR : quantity = 0 , book out of stock');
end if;
end;

/*update tables when deleting the order*/
create or replace trigger update_deleted_order
after delete on cust_order
for each row
BEGIN
update book set quantity = quantity +1 where book_id =:old.book_id;
```

```
delete from order_history where order_id = :old.order_id;
end;

/**PACKAGE*/

create or replace PACKAGE bookstore_package
IS
procedure list_customer_orders;
function printf_nb_book_per_gender(u_gender customer.gender%type) return VARCHAR2;
procedure nb_book_sold_per_quarter;
procedure revenue_per_quarter;
procedure nb_oders_per_status;
PROCEDURE nb_books_buought_by_country;
FUNCTION nb_order_in_period(d_start DATE , d_end DATE) return VARCHAR2;
FUNCTION total_spending_per_customer(c_id cust_order.customer_id%type) return VARCHAR2;
PROCEDURE top_buyers;
PROCEDURE nb_copy_sold_per_book;
PROCEDURE nb_books_per_method;
PROCEDURE best_seller_book;
end bookstore_package;

/*PACKAGE BODY*/
create or replace PACKAGE BODY bookstore_package
```

```
IS
---- PROCEDURE 1 ------
   procedure list_customer_orders
      IS
      cursor c1 is select cust_order.order_id , first_name , last_name from customer , cust_order where customer.customer_id = cust_order.customer_id;
      rec_c c1%rowtype;
      begin
      open c1;
      loop
      fetch c1 into rec_c;
      exit when c1%notfound;
      DBMS_OUTPUT.PUT_LINE('Order Id: '||rec_c.order_id|| ', First name: '||rec_c.first_name||', Last name: '||rec_c.last_name);
      end loop;
      EXCEPTION
      WHEN OTHERS THEN
      dbms_output.put_line(sqlcode);
      dbms_output.put_line(sqlerrm);
   end;


---- PROCEDURE 2 ------
   function printf_nb_book_per_gender(u_gender customer.gender%type)
      return VARCHAR2
      as
      nb_book number(3);
```

```
        n_result varchar(1000);

        begin

        DBMS_OUTPUT.PUT_LINE('********************NUMBER OF BOOKS BOUGHT PER GENDER***************** ');

        select COUNT(book.title) into nb_book from book , cust_order , customer

        where book.book_id = cust_order.book_id AND customer.customer_id = cust_order.customer_id and customer.gender = u_gender;

        n_result := 'Number of books bought by '||u_gender||' = '||nb_book;

        return n_result;

    end printf_nb_book_per_gender;


---- PROCEDURE 3------

    procedure nb_book_sold_per_quarter

        as

        cursor c1 is select CUST_ORDER.order_date as od , count(book.book_id) as nb_book from CUST_ORDER , book , order_history where

        CUST_ORDER.book_id = book.book_id and order_history.status_id = 3 group by CUST_ORDER.order_date ;

        rec_c c1%rowtype;

        first_q NUMBER(2);

        second_q NUMBER(2);

        fourth_q NUMBER(2);

        third_q NUMBER(2);

        begin

        first_q := 0;

        second_q :=0;
        third_q :=0;

        fourth_q := 0;

        open c1;

        loop

        fetch c1 into rec_c;

        exit when c1%notfound;

        IF rec_c.od BETWEEN to_date('01/01/2021','mm/dd/yyyy') AND
to_date('03/31/2021','mm/dd/yyyy') THEN

        first_q := first_q + 1;

        elsif rec_c.od BETWEEN to_date('04/01/2021','mm/dd/yyyy') AND
to_date('06/30/2021','mm/dd/yyyy') THEN

        second_q := second_q +1;

        elsif rec_c.od BETWEEN to_date('07/01/2021','mm/dd/yyyy') AND
to_date('09/30/2021','mm/dd/yyyy') THEN

        third_q := third_q +1;

        elsif rec_c.od BETWEEN to_date('10/01/2021','mm/dd/yyyy') AND
to_date('12/31/2021','mm/dd/yyyy') THEN

        fourth_q := fourth_q +1;

        END IF;

        end loop;

        close c1;

        dbms_output.put_line('NUMBER OF BOOKS BOUGHT IN THE FIRST
QUARTER');

        dbms_output.put_line(first_q);

        dbms_output.put_line('NUMBER OF BOOKS BOUGHT IN THE SECOND
QUARTER');

        dbms_output.put_line(second_q);
```

```
        dbms_output.put_line('NUMBER OF BOOKS BOUGHT IN THE THIRD QUARTER');
        dbms_output.put_line(third_q);
        dbms_output.put_line('NUMBER OF BOOKS BOUGHT IN THE FOURTH QUARTER');
        dbms_output.put_line(fourth_q);
        EXCEPTION
        WHEN OTHERS THEN
        dbms_output.put_line(sqlcode);
        dbms_output.put_line(sqlerrm);
    end nb_book_sold_per_quarter;

---- PROCEDURE 4 ------
    procedure revenue_per_quarter
        as
        cursor c1 is select CUST_ORDER.order_date as od , cust_order.total_price as total_sum from CUST_ORDER , order_history where
        cust_order.order_id = order_history.order_id and order_history.status_id = 3;
        rec_c c1%rowtype;
        first_q_revenue DECIMAL(10, 2);
        second_q_revenue DECIMAL(10, 2);
        fourth_q_revenue DECIMAL(10, 2);
        third_q_revenue DECIMAL(10, 2);
        begin
        first_q_revenue := 0;
```

```
second_q_revenue :=0;
third_q_revenue :=0;
fourth_q_revenue := 0;
open c1;
loop
fetch c1 into rec_c;
exit when c1%notfound;
IF rec_c.od BETWEEN to_date('01/01/2021','dd/mm/yyyy') AND to_date('31/03/2021','dd/mm/yyyy') THEN
first_q_revenue := first_q_revenue + rec_c.total_sum;
elsif rec_c.od BETWEEN to_date('01/04/2021','dd/mm/yyyy') AND to_date('30/06/2021','dd/mm/yyyy') THEN
second_q_revenue := second_q_revenue + rec_c.total_sum;
elsif rec_c.od BETWEEN to_date('01/07/2021','dd/mm/yyyy') AND to_date('30/09/2021','dd/mm/yyyy') THEN
third_q_revenue := third_q_revenue + rec_c.total_sum;
elsif rec_c.od BETWEEN to_date('01/10/2021','dd/mm/yyyy') AND to_date('31/12/2021','dd/mm/yyyy') THEN
fourth_q_revenue := fourth_q_revenue + rec_c.total_sum;
END IF;
end loop;
close c1;
dbms_output.put_line('REVENUES OF THE FIRST QUARTER');
dbms_output.put_line(first_q_revenue||'$');
dbms_output.put_line('REVENUES OF THE SECOND QUARTER');
dbms_output.put_line(second_q_revenue||'$');
```

```
        dbms_output.put_line('REVENUES OF THE THIRD QUARTER');
        dbms_output.put_line(third_q_revenue||'$');
        dbms_output.put_line('REVENUES OF THE FOURTH QUARTER');
        dbms_output.put_line(fourth_q_revenue||'$');
        EXCEPTION
        WHEN OTHERS THEN
        dbms_output.put_line(sqlcode);
        dbms_output.put_line(sqlerrm);
    end revenue_per_quarter;



---- PROCEDURE 5 ------
    procedure nb_oders_per_status
        as
        nb_orders NUMBER(3);
        statusValue order_status.status_value%type;
        cursor c1 is select order_status.status_value
,count(order_history.status_id) from order_history , order_status
        where order_history.status_id = order_status.status_id group by
order_status.status_value;


        begin
        open c1;
        loop
        fetch c1 into statusValue , nb_orders;
        exit when c1%notfound;
        DBMS_OUTPUT.PUT_LINE('Number of book of status [ '||statusValue||' ] = '||nb_orders);
        end loop;
        close c1;
        EXCEPTION
        WHEN OTHERS THEN
        dbms_output.put_line(sqlcode);
        dbms_output.put_line(sqlerrm);
    end nb_oders_per_status;
---- PROCEDURE 6 ------
    PROCEDURE nb_books_buought_by_country
        as
        cursor c1 is select customer.country as country , count(cust_order.order_id) as nb_book from cust_order , customer
        where customer.customer_id = cust_order.customer_id group by customer.country ORDER BY nb_book DESC;
        rec_c c1%rowtype;
        begin
        DBMS_OUTPUT.PUT_LINE('*************NUMBER OF BOOKS BOUGHT BY EVERY COUNTRY*****************');
        open c1;
        loop
        fetch c1 into rec_c;
        exit when c1%notfound;
        DBMS_OUTPUT.PUT_LINE('COUNTRY NAME : '||rec_c.country||' => NUMBER OF BOOKS BOUGHT : '||rec_c.nb_book);
        end loop;
```

```
      close c1;
      EXCEPTION
      WHEN OTHERS THEN
      dbms_output.put_line(sqlcode);
      dbms_output.put_line(sqlerrm);
   end nb_books_buought_by_country;
---- PROCEDURE 7 ------
   FUNCTION nb_order_in_period(d_start DATE , d_end DATE)
      return VARCHAR2
      AS
      nb NUMBER(2);
      result varchar(1200);
      begin
      select count(cust_order.order_id) into nb from cust_order where cust_order.order_date BETWEEN d_start and d_end;
      result := 'NUMBER OF ORDER BETWEEN '||d_start||' AND '||d_end||' : '||nb;
      return result;
   end nb_order_in_period;
---- PROCEDURE 8 ------
   FUNCTION total_spending_per_customer(c_id cust_order.customer_id%type)
      return VARCHAR2
      AS
      result varchar2(1200);
      begin
```

```
        DBMS_OUTPUT.PUT_LINE('***********TOTAL SPENDING PER CUSTOMER**************');
        for cur_var in (
        select first_name , last_name , sum(total_price) as total_spending from cust_order , customer
        where cust_order.customer_id = c_id and cust_order.customer_id = customer.customer_id group by first_name,last_name)
        loop
        result := cur_var.first_name||' '||cur_var.last_name||' has spent: '||cur_var.total_spending||'$';
        end loop;
        return result;
    end total_spending_per_customer;
---- PROCEDURE 9 ------
    PROCEDURE top_buyers
        IS
        begin
        DBMS_OUTPUT.PUT_LINE('LIST OF TOP BUYERS');
        for cur_var in (select customer.first_name as fn , count(cust_order.order_id) as nb_order from customer , cust_order where customer.customer_id = cust_order.customer_id
        group by customer.first_name ORDER BY count(cust_order.order_id) DESC)
        loop
        DBMS_OUTPUT.PUT_LINE(cur_var.fn || '      '||cur_var.nb_order);
        end loop;
        EXCEPTION
    WHEN OTHERS THEN
    dbms_output.put_line(sqlcode);
    dbms_output.put_line(sqlerrm);
  end;
---- PROCEDURE 10 ------
      PROCEDURE nb_copy_sold_per_book
            AS
            begin
            for c1 in (select book.title as title , count(cust_order.order_id) as
nb from cust_order , book
            where cust_order.book_id = book.book_id group by book.title)
            loop
            dbms_output.put_line('Book Title: '||c1.title||' =>     Number of
Copies: '||c1.nb);
            end loop;
            EXCEPTION
            WHEN OTHERS THEN
            dbms_output.put_line(sqlcode);
            dbms_output.put_line(sqlerrm);
      end nb_copy_sold_per_book;


---- PROCEDURE 11 ------
      PROCEDURE nb_books_per_method
            AS
```

```
                cursor c1 is select shipping_method.method_name as
shipping_name ,count(cust_order.order_id) as nb_order from cust_order ,
shipping_method

                where cust_order.method_id = shipping_method.method_id group
by shipping_method.method_name;

                rec_c c1%rowtype;

                begin

                open c1;

                loop

                fetch c1 into rec_c;

                exit when c1%notfound;

                DBMS_OUTPUT.PUT_LINE(rec_c.nb_order||' Order has been
shipping with '||rec_c.shipping_name);

                end loop;

                close c1;

                EXCEPTION

                WHEN OTHERS THEN

                dbms_output.put_line(sqlcode);

                dbms_output.put_line(sqlerrm);

        end nb_books_per_method;

---- PROCEDURE 12 ------

   PROCEDURE best_seller_book

      IS

      cursor c1 is select book.title, author.author_name ,
count(cust_order.book_id) as nb_of_books from cust_order , book,author

      where cust_order.book_id = book.book_id AND author.author_id =
book.author_id
```

```
        group by book.title,author.author_name ORDER BY nb_of_books DESC;
        c_rec c1%rowtype;
        begin
        open c1;
        loop
        fetch c1 into c_rec;
        exit when c1%rowcount=2;
        dbms_output.put_line('*****************BEST SELLER
BOOK*******************');
        dbms_output.put_line('Title : '||c_rec.title);
        dbms_output.put_line('Number of sales: '||c_rec.nb_of_books);
        dbms_output.put_line('Author: '||c_rec.author_name);
        end loop;
        close c1;
        EXCEPTION
        WHEN OTHERS THEN
        dbms_output.put_line(sqlcode);
        dbms_output.put_line(sqlerrm);
    end;
end bookstore_package;
```