**HIGHER DISTINCTION ASSIGNMENT PROJECT**

1. **A simple query of a single table:**

List Book IDs, Book Names, Price in AUD, Genre IDs, Author IDs from the book\_collection table, who has a price greater than or equal to 25 and whose stock is more than or equal to 20. Finally, order the result by Book Name in ascending order.

**Query Output Table:**

Table

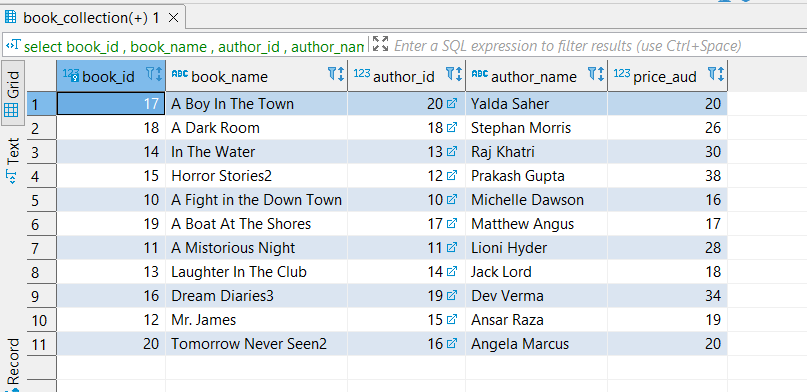
Description automatically generated

1. **A query which uses natural join**

List all book IDs, Book Names, Author IDs, Author Names, their price in AUD, from Book Collection table using natural join, whose Author's ID is between

10 and 20 and order the result by Author Name in descending order.

**Output Query Table:**

****

List Book IDs, Book Names, Author IDs, Author Names using the cross product and list all the authors whose name starts with M and then order the query by price in ascending order.

**3.The cross product equivalent to the "natural join" query above.**

**Query Output Table:**

Table

Description automatically generated

**4.A query involving a “Group by”, perhaps also with a “HAVING”.**

List Purchase IDs, order date, price, discount, net price of orders(net price = price\* quantity \*(1-discount)), employee names from tables Purchase Information. Employee Information.

Use group by along with having condition. Display the above list whose order month is august and order by order date in ascending order.

**Query Output Table:**

Graphical user interface

Description automatically generated

**5. A query which uses a sub query.**

List all the Book names, Prices of books, and Genre IDs from the Book Collection table. It contains a sub-query that filters and extracts books from the list of books with minimum price and these books must belong to Genre Id 5.

**Query Output Table:**

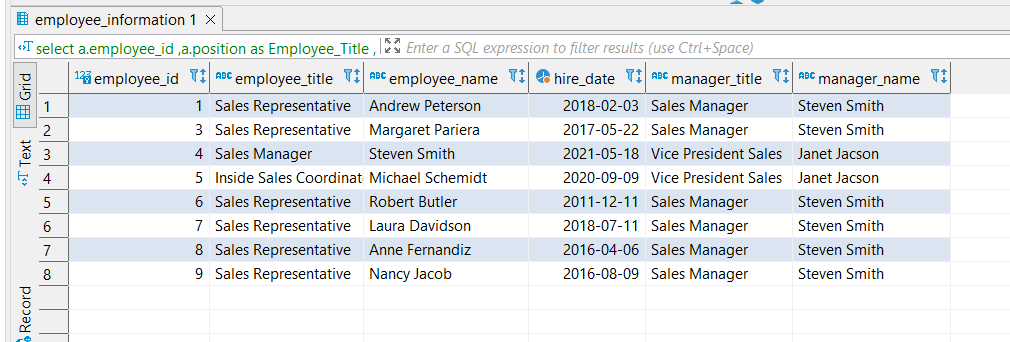
**Table

Description automatically generated**

**6.A cross product which cannot be implemented using the words “natural join” (e.g. self join).**

List all the employees, their position, date of hire and these employees should report to manager from Employee Information table using self join.

**Query Output Table:**

****

**ANSWERS**

1. **A simple query of a single table:**

**Query:**

select Book\_Id, Book\_Name

, price\_aud,Genre\_Id

, Author\_Id

from book\_collection

where in\_stock >=20 and price\_aud >=25

order by book\_name asc;

**2.A query which uses natural join**

**Query:**

select book\_id

, book\_name

, author\_id

, author\_name

, price\_aud

from book\_collection natural join author\_details

where author\_id between 10 and 20

order by author\_name desc;

**3.The cross product equivalent to the "natural join" query above.**

**Query:**

**select** Book\_Collection.book\_id

, Book\_Collection.book\_name

, Author\_Details.author\_id

, Author\_Details.author\_name

,Book\_Collection.price\_aud

**from** Book\_Collection, Author\_Details

**where** Author\_Details.author\_id = Book\_Collection.author\_id

**and** author\_name **like** 'M%'

**order** **by** price\_aud **asc**;

**4.A query involving a “Group by”, perhaps also with a “HAVING”.**

**Query:**

select Purchase\_Information.purchase\_id

,Purchase\_Information.Date\_Order

,Purchase\_Information.price

,Purchase\_Information.quantity

,Purchase\_Information.discount

,Purchase\_Information.quantity \*(1-Purchase\_Information.discount)\*Purchase\_Information.price as Net\_Price

,Employee\_Information.employee\_name

from Purchase\_Information inner join Employee\_Information on Purchase\_Information.employee\_id = Employee\_Information.employee\_id

group by Purchase\_Information.purchase\_id ,Employee\_Information.employee\_name

having Extract('month' from date\_order::date) = 08

order by purchase\_Information.Date\_Order asc;

**5. A query which uses a sub query.**

**Query:**

select book\_name

, price\_aud

, genre\_id

from book\_collection

where genre\_id =5

and price\_aud >=(

select min(price\_aud)

from book\_collection

where genre\_id =5);

**6. A cross product which cannot be implemented using the words “natural join” (e.g. self join)**

**Query:**

select a.employee\_id

,a.position as Employee\_Title

,a.employee\_name as Employee\_Name

,a.Date\_Hire as Hire\_Date

, b.position as Manager\_Title

, b.employee\_name as Manager\_Name

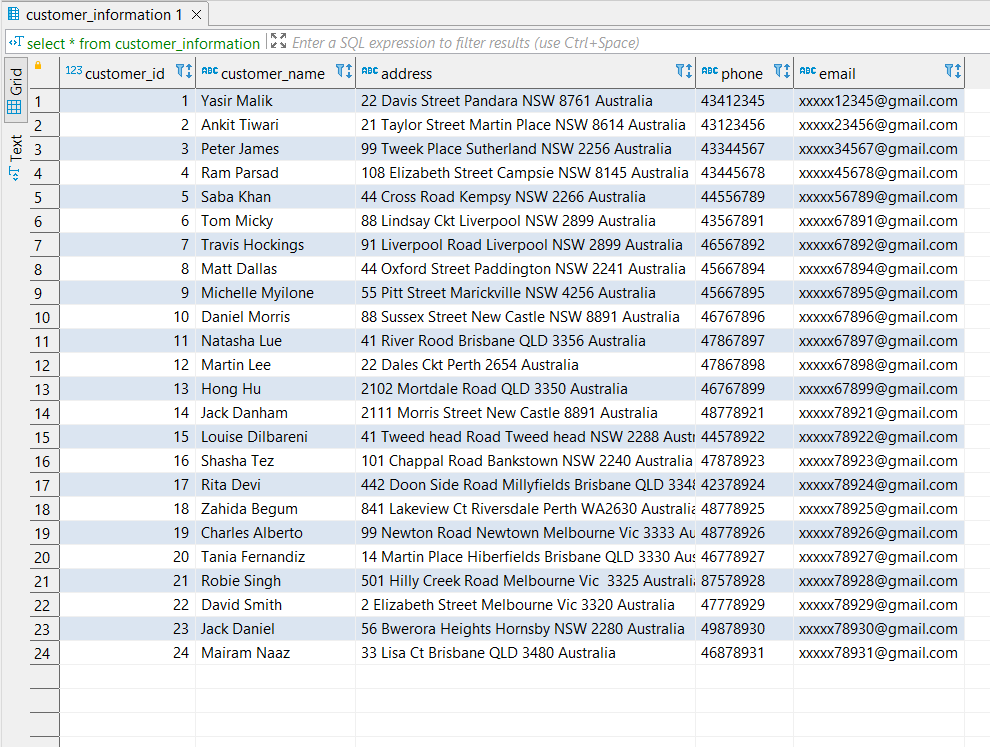
from employee\_information as a , employee\_information as b

where a.report\_to\_manager=b.employee\_id;

**ADDITIONAL TASK**

**1.CUSTOMER\_INFORMATION QUERY AND TABLE**

select \* from customer\_information;



**2.QUERY TO INSERT A NEW ROW ENTRY INTO CUSTOMER\_INFORMATION TABLE.**

**Query:**

insert into Customer\_Information values(25, 'Teagen Lioni', '65 Lakemba Ct Sydney NSW 2280 Australia', 0468789332, 'xxxxx93321@gmail.com');

**TABLE OUTPUT:**

Graphical user interface, text, application

Description automatically generated

**3.QUERY TO UPDATE PHONE NUMBER IN CUSTOMER INFORMATION TABLE.**

**Query:**

update customer\_information set phone=0468789112 where customer\_id=25;

**OUTPUT:**

Text, table

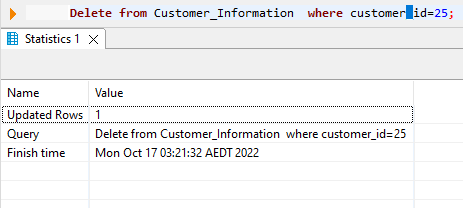
Description automatically generated

**4.DELETE STATEMENT TO DELETE AN ENTRY FROM TABLE.**

**Query:**

Delete from Customer\_Information where customer\_id=25;

**OUTPUT:**



**ON DELETE CASCADE QUERY**

**5.APPLYING ON DELETE CASCADE CONSTRAINT ON BOOK\_GENRES**

**QUERY**

**create** **table** Book\_Collection

(

Book\_Id **smallint**,

Book\_Name **character** **varying** **not** **null**,

Genre\_Id **integer**,

Author\_Id **integer**,

Publisher\_Id **integer**,

Price\_AUD **integer**,

In\_Stock **integer**,

**constraint** Book\_Collectionpk **primary** **key** (Book\_Id),

**constraint Book\_Collectionfk\_Book\_Genres foreign key (Genre\_Id) references Book\_Genres on delete cascade,**

**constraint** Book\_Collectionfk\_Author\_Details **foreign** **key** (Author\_Id) **references** Author\_Details,

**constraint** Book\_Collectionfk\_Book\_Publisher **foreign** **key** (Publisher\_Id) **references** Book\_Publisher,

**constraint** unq\_Book\_Collection\_book\_name **unique** (book\_name)

);

**OUTPUT TABLE OF BOOK\_COLLECTION**

**Graphical user interface, table

Description automatically generated**

**(5.1).QUERY TO EXECUTE ON DELETE CASCADE**

**Comment: Book\_Genre is Parent here and Book\_Collection is Child**

**Query:**

**delete** **from** book\_genres **where** genre\_id  =1;

**OUTPUT FOR ON DELETE CASCADE ON BOOK\_GENRE**

Graphical user interface, text, application, email

Description automatically generated

**6.APPLYING ON DELETE RESTRICT ON CUSTOMER\_INFORMATION**

**Query:**

**create** **table** Purchase\_Information

(

Purchase\_Id **integer**,

Book\_Id **integer**,

Price **integer**,

Quantity **integer**,

Discount **float**,

Customer\_Id **integer**,

Employee\_Id **integer**,

Date\_Order **Date**,

Date\_Shipped **character** **varying**,

Shipping\_Address **character** **varying**,

**constraint** Purchase\_Informationpk **primary** **key** (Purchase\_Id),

**constraint** Purchase\_Informationfk\_Book\_Collection **foreign** **key** (Book\_Id) **references** Book\_Collection,

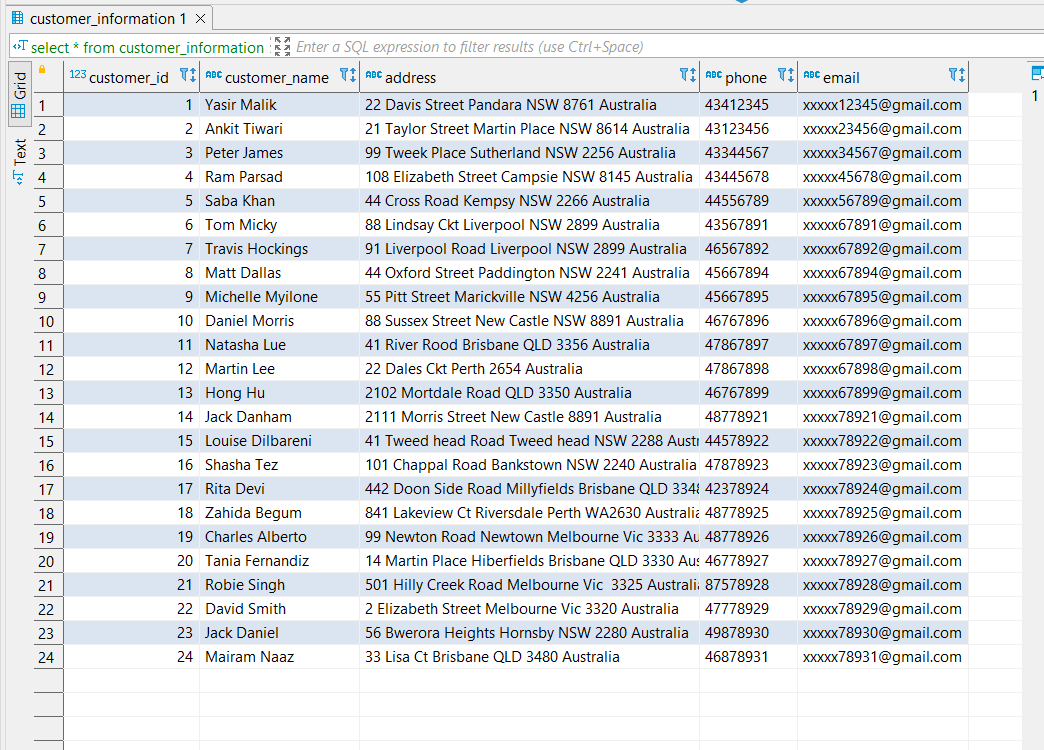
**constraint** Purchase\_Information\_Price **check** (Price > 0),

**constraint** Book\_Purchasefk\_Customer\_Information **foreign** **key** (Customer\_Id) **references** Customer\_Information **on** **delete** **restrict**,

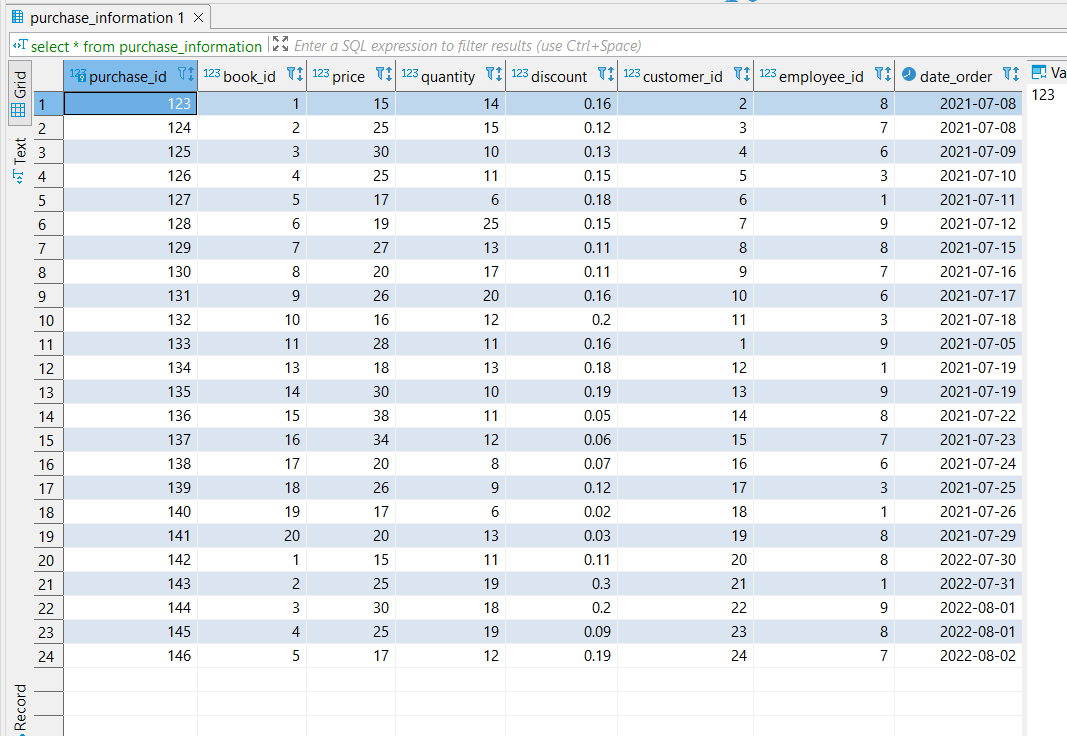
**constraint** Book\_Purchasefk\_Employee\_Information **foreign** **key** (Employee\_Id) **references** Employee\_Information

);

**OUTPUT TABLE FOR CUSTOMER\_INFORMATION**

****

**OUTPUT TABLE FOR PURCHASE\_INFORMATION**

****

**(6.1).QUERY OF ON DELETE RESTRICT ON CUSTOMER\_INFORMATION**

**QUERY:**

**delete** **from** customer\_information

**where** customer\_id =24;

**OUTPUT OF ON DELETE RESTRICT ON CUSTOMER\_INFORMATION**

Graphical user interface, text, application, email

Description automatically generated

**REFERENCES :**

**1.** [Booktopia - Books, Online Books, #1 Australian online bookstore, Buy Discount Books, eBooks and DVDs from Australia and the world.](https://www.booktopia.com.au/?msclkid=f65fe960552c172165c58a2881bd3795&utm_source=bing&utm_medium=cpc&utm_campaign=Booktopia%20-%20AU%20-%20Search%20-%20Competitor%20(Exact,%20Phrase%20%2B%20Broad)&utm_term=amazons%20books&utm_content=Amazon)

**2.** [Book Depository: Free delivery worldwide on over 20 million books](https://www.bookdepository.com/?utm_source=Affiliate_AU&utm_medium=Affiliate_Marketing&utm_campaign=177157%20&utm_term=httpwwwfindercomau&awc=10920_1665719740_da97de068378e01b3b42330b853e9fe2)

**3.** [Books | Online Bookshop | QBD](https://www.qbd.com.au/?cfclick=ddaeecbc078044dfbe6c571fab2b4e29)