**User Manual**

A close up of text on a white background

Description automatically generated

Queries

1. Find the titles of all books by Pratchett that cost less than $10

Return a table that contains the title of that books written by Pratchett and cost less than $10



SELECT Title

FROM BOOK AS B

WHERE B.Price < 10 AND B.Author = 'Pratchett';

1. Give all the titles and their dates of purchase made by a single customer

Return a table that contains the title and date of the purchases made by a single customer

A picture containing knife

Description automatically generated

SELECT B.Title, P.Date

FROM BOOK AS B, PURCHASE AS P, C AS CUSTOMER

WHERE B.Isbn = P.Book AND P.Customer = C.Email;

1. Find the titles and ISBNs for all books with less than 5 copies in stock

Return a table that contains the title and ISBN for all books with less than 5 copies in inventory

A picture containing knife

Description automatically generated

SELECT B.Title, B.Isbn

FROM BOOK AS B, INVENTORY AS I

WHERE B.Isbn = I.Isbn

GROUP BY B.Title, B.Isbn

HAVING sum(I.Quantity) < 5;

1. Give all the customers who purchased a book by Pratchett and the titles of Pratchett books they purchased

Return a table that contains the customers who purchased a book by Pratchett and the titles of Pratchett books they purchased

A picture containing knife, table

Description automatically generated

SELECT C.Email, C.Name, B.Title

FROM CUSTOMER AS C, BOOK AS B, PUBLISHES AS P, PURCHASE AS PUR

WHERE P.Isbn = B.Isbn AND P.Author = 'Pratchett' AND PUR.Book = B.Isbn;

1. Find the total number of books purchased by a single customer

Return a table that contains the total number of books a customer has purchased and the email of the customer

A picture containing knife

Description automatically generated

SELECT P.Customer, COUNT(Book)

FROM PURCHASE AS P, CUSTOMER AS C

WHERE P.Customer = C.Email;

1. Find the customer who has purchased the most books and the total number of books they have purchased

Return a table that contains the customer who has purchased the most books and the amount of the books they have purchased

A picture containing knife

Description automatically generated

SELECT L.Customer, L.Num

FROM PURCHASE AS P, CUSTOMER AS C,

(SELECT P.Customer AS Customer, COUNT(Book) AS Num

FROM PURCHASE AS P, CUSTOMER AS C

WHERE P.Customer = C.Email) AS L

WHERE L.Num = MAX(L.Num);

1. Find the CUSTOMER with the most Reward\_point on his/her account

Return a table that contains the email and name of the customer who has the most reward point on his or her account

A picture containing knife

Description automatically generated

SELECT Email, Name

FROM CUSTOMR C, ACCOUNT A

WHERE C.Email = A.Email

GROUP BY C.Email, C.Name

HAVING A.Reward\_point = MAX(A.Reward\_point)

1. Find the most expensive BOOK with all the DISCOUNT applied

Return a table that contains ISBN and title of books that has the most expensive price with applied discount

A picture containing knife, table

Description automatically generated

SELECT Isbn, Title

FROM (BOOK LEFT OUTER JOIN DISCOUNT)

WHERE Percentage IS NOT NULL

GROUP BY Isbn, Title

HAVING Price\*Percentage = MAX(Price\*Percentage);

1. Find the total price of all the BOOK for each stock (quantity \* price)

Return a table that contains the total price of all the books and their ISBN for each stock

A picture containing knife

Description automatically generated

SELECT Isbn, Quantity\*Price

FROM BOOK NATURAL JOIN INVENTORY

GROUP BY Isbn

HAVING Quantity\*Price=SUM(Quantity);

1. Provide a list of customer names, along with the total dollar amount each customer has spent.

Return a table that contains the names of the customers and the total dollar amount each customer has spent



SELECT C.Name, SUM(P.Actual\_Price)

FROM PURCHASE AS P, CUSTOMER AS C

WHERE P.Customer = C.Email

GROUP BY P.Customer

1. Provide a list of customer names and e-mail addresses for customers who have spent more than the average customer.

Return a table that contains the names and e-mail addresses of the customers who have spent more than the average customer



SELECT Name, Email

FROM CUSTOMER NATURAL JOIN (SELECT SUM Actural\_Price AS Total\_amount

FROM PURCHASE AS P

GROUP BY P.Customer ) AS TOTAL

HAVING TOTAL.Toal\_amount > AVG (TOTAL.Total\_amount)

1. Provide a list of the titles in the database and associated total copies sold to customers, sorted from the title that has sold the most individual copies to the title that has sold the least.

Return a table that contains the total number of each book that has been sold to customers, and the titles of those books, sorted by the number of books in descending number.

A picture containing knife, table

Description automatically generated

SELECT B.Title, COUNT(P.Book)

FROM PURCHASE AS P, BOOK AS B

WHERE P.Book = B.Isbn

GROUP BY P.Book

ORDER BY COUNT(P.Book) DESC

1. Provide a list of the titles in the database and associated dollar totals for copies sold to customers, sorted from the title that has sold the highest dollar amount to the title that has sold the smallest.

Return a table that contains the total dollar amount of each book that has been sold to customers, and the titles of those books, sorted by the dollar amount in descending order.

A picture containing knife, table

Description automatically generated

SELECT B.Title, P.Actual\_Price \* COUNT(P.Book) AS Total\_dollars

FROM PURCHASE AS P, BOOK AS B

WHERE P.Book = B.Isbn

GROUP BY P.Book

ORDER BY P.Actual\_Price \* COUNT(P.Book) DESC

1. Find the most popular author in the database (i.e. the one who has sold the most books)

Return a table that contains the name of the author who sold the most books

A picture containing knife, table

Description automatically generated

SELECT A.Name, MAX(Total)

FROM AUTHOR AS A, (SELECT A.AuthorId, COUNT(P.Book) AS Total

FROM PURCHASE AS P, BOOK AS B, WRITES AS W

WHERE P.Book = B.Isbn AND B.Isbn = W.Isbn

GROUP BY W.AuthorId) AS R1

WHERE R1.AuthorId = A.AuthorId

GROUP BY A.AuthorId, A.Name

1. Find the most profitable author in the database for this store (i.e. the one who has brought in the most money)

Return a table that contains the name of the author who brought in the most money and the dollar amount

A screenshot of a cell phone

Description automatically generated

SELECT A.Name, MAX(Total)

FROM AUTHOR AS A, (SELECT A.AuthorId, P.Actual\_Price \* COUNT(P.Book) AS Total

FROM PURCHASE AS P, BOOK AS B, WRITES AS W

WHERE P.Book = B.Isbn AND B.Isbn = W.Isbn

GROUP BY W.AuthorId) AS R1

WHERE R1.AuthorId = A.AuthorId

GROUP BY A.AuthorId, A.Name

1. Provide a list of customer information for customers who purchased anything written by the most profitable author in the database.

Return a table that contains the information of customers who purchased anything written by the most profitable author

A screenshot of a cell phone

Description automatically generated

SELECT C.Name, C.Email, C.Phone, C.Address

FROM (SELECT A.AuthorId, MAX(Total)

FROM (SELECT A.AuthorId, P.Actual\_Price \* COUNT(P.Book) AS Total

FROM PURCHASE AS P, BOOK AS B, WRITES AS W

WHERE P.Book = B.Isbn AND B.Isbn = W.Isbn

GROUP BY W.AuthorId) AS R1,

AUTHOR AS A

WHERE R1.AuthorId = A.AuthorId

GROUP BY A.AuthorId) AS R2, CUSTOMER AS C

WHERE P.Book = B.Isbn AND B.Isbn = W.Isbn AND W.AuthorId = R2.AuthorId

1. Provide the list of authors who wrote the books purchased by the customers who have spent more than the average customer.

Return a table that contains the name of the authors whose books are purchased by the customers who have spent more than average customer

A picture containing knife

Description automatically generated

SELECT A.Name

FROM AUTHOR AS A, WRITES AS W, PURCHASE AS P, CUSTOMER AS C,

(SELECT C1.Email AS Email, sum(P1.Actual\_Price) AS Spent

FROM PURCHASE P1, CUSTOMER C1

WHERE P1.Customer = C1.Email

GROUP BY P1.Customer) AS personal\_sum

WHERE A.AuthorId = W.AuthorId AND W.Isbn = B.Isbn AND C.Email = personal\_sum.Email AND personal\_sum.Spent > avg(personal\_sum.Spent) AND P.Customer = C.Email AND P.Book = B.Isbn

Insert syntax for adding books, publisher, authors and customers

1. Add books

/\*Since each book must have publisher and author, and each author, publisher must write/publish a book the book insert will followed by writes and publishes relation insert, then followed by author and publisher insert, vise versa\*/

INSERT INTO BOOK(ISBN, Title, Year, Price, Category)

VALUES ('0782140661','OCP:Oracle9iCertificationKit',2002,104.97,'Computer');

INSERT INTO WRITES(ISBN,AuthorId)

VALUES ('0782140661',12345)

INSERT INTO AUTHOR(AuthorId,name)

VALUES (12345, 'ChipDawes');

INSERT INTO PUBLISHES(ISBN,Publisher)

VALUES ('0782140661', 'Sybex');

INSERT INTO PUBLISHER(Name,Phone,Address)

VALUES ('Sybex','1234567890','this is an address');

1. Add publisher

INSERT INTO PUBLISHER(Name,Phone,Address)

VALUES ('Sybex','1234567890','this is an address');

INSERT INTO PUBLISHES(ISBN,Publisher)

VALUES ('0782140661', 'Sybex');

INSERT INTO BOOK(ISBN, Title, Year, Price, Category)

VALUES ('0782140661','OCP:Oracle9iCertificationKit',2002,104.97,'Computer');

INSERT INTO WRITES(ISBN,AuthorId)

VALUES ('0782140661',12345)

INSERT INTO AUTHOR(AuthorId,name)

VALUES (12345, 'ChipDawes');

1. Add authors

INSERT INTO AUTHOR(AuthorId,name)

VALUES (12345, 'ChipDawes');

INSERT INTO WRITES(ISBN,AuthorId)

VALUES ('0782140661',12345)

INSERT INTO BOOK(ISBN, Title, Year, Price, Category)

VALUES ('0782140661','OCP:Oracle9iCertificationKit',2002,104.97,'Computer');

INSERT INTO PUBLISHES(ISBN,Publisher)

VALUES ('0782140661', 'Sybex');

INSERT INTO PUBLISHER(Name,Phone,Address)

VALUES ('Sybex','1234567890','this is an address');

1. Add customer

/\*A customer can have 0 or 1 account, and make 0 or more purchases, so no dependencies\*/

INSERT INTO CUSTOMER(Email,Name,Phone,Address)

VALUES ('email@email.com','A Name','0123456789', 'This is an address');

Delete syntax for removing books, publisher, authors and customers

1. Remove books

/\*When delete a book, delete its publisher and writer, if no other book written/published by same ones\*/

DELETE FROM BOOK

WHERE ISBN = '0782140661';

DELETE FROM AUTHOR

WHERE (SELECT ISBN

FROM (AUTHOR LEFT OUTER JOIN WIRTES)

WHERE ISBN = '0782140661' )

DELETE FROM WRITES

WHERE ISBN = '0782140661';

DELETE FROM PUBLISHER

WHERE (SELECT ISBN

FROM (PUBLISHER LEFT OUTER JOIN PUBLISHES)

WHERE ISBN = '0782140661' )

DELETE FROM PUBLISHES

WHERE ISBN = '0782140661';

1. Remove publisher

/\*When deleting publisher, delete all books publishes by them, and the books’ author\*/

DELETE FROM PUBLISHER

WHERE Name = 'Sybex';

DELETE FROM AUTHOR

WHERE (SELECT Authorid

FROM WRITES

WHERE (SELECT ISBN

FROM PUBLISHES

WHERE Publisher = 'Sybex'));

DELETE FROM WRITES

WHERE (SELECT ISBN

FROM PUBLISHES

WHERE Publisher = 'Sybex');

DELETE FORM BOOK

WHERE (SELECT ISBN

FROM (PUBLISHERES RIGHT OUTER JOIN BOOK)

WHERE PUBLISHER.Publisher = 'Sybex');

DELETE FROM PUBLISHES

WHERE Publisher = 'Sybex';

1. Remove authors

/\*When deleting author, delete all books written by them(if they are the only author), and the books’ publisher\*/

DELETE FROM AUTHOR

WHERE AuthoreId = '12345';

DELETE FROM PUBLISHER

WHERE (SELECT Publisher

FROM PUBLISHES

WHERE (SELECT ISBN

FROM WRITES

WHERE AuthorId='12345'));

DELETE FROM PUBLISHES

WHERE (SELECT ISBN

FROM WRITES

WHERE AuthorId='12345');

DELETE FORM BOOK

WHERE (SELECT ISBN

FROM (WRITES RIGHT OUTER JOIN BOOK)

WHERE AuthorId='12345');

DELETE FROM WRITES

WHERE AuthorId='12345';

1. Remove customer

/\*Remove the account also, if the customer has one\*/

DELETE FROM ACCOUNT

WHERE Email = ‘email@email.com’

DELETE FROM CUSTOMER

WHERE Email = ‘email@email.com’