



UNIVERSITY OF
PORTSMOUTH

DATABASE MANAGEMENT.

COURSEWORK 2: Folly Bee Pubs Database
Group work

Database Solution for The Folly Bee Pubs

GROUP 2

School of Computing, University of Portsmouth– Portsmouth

M30208: Dr Olumuyiwa Matthew

Date: 21st January 2022

GROUP Members.

STUDENT ID

UP2076679

UP2076679

UP2076679

UP2076679

UP2076679

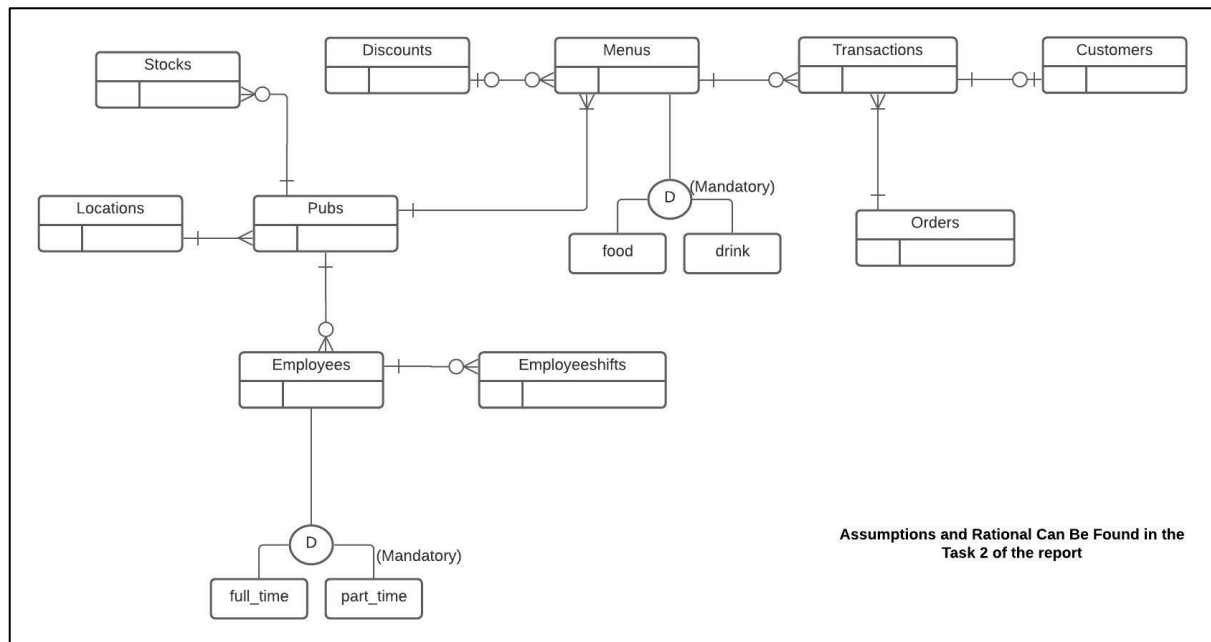
CONFIDENTIAL

Table of Contents

<i>Database Solution for the Folly Bee Pubs</i>	2
Task T1: EERD	4
Task T2: Rationale and Assumptions	4
Task T3: Data Dictionary/Scripts	5
Data Dictionary.....	5
Scripts To Create Tables	9
Task T4: SQL Queries/ Final Demonstration	13
SQL Queries	13
Demonstration	15
References:	18
Appendix	19

Task T1: EERD

Figure 1: This EERD diagram shows the entity relations between different tables of folly bees database diagram.



Generated using Lucid chart

Description: Above figure represent EERD diagram of the Folly Bees Pubs total 10 tables was created for the purpose of this database. Some assumption was made during the creation of this EERD as mentioned in Task 2.

Task T2: Rationale and Assumptions

Rationale and Assumptions

1. When any pub has been sold out one of the items, they can't update stock on that item or dish name they can just update the items which are used to create that dish (item).

Example: if the toast is out of stock in any of the venue, venue manager can update the shortage of ingredients which are used to make toast, similarly this will apply to other dishes. So in the stocks table we have considered the quantity column.

2. To know whether the particular employee is on the shift or not we have used on_shift field in the employees shift tables which is a boolean value when it is true it means that employee is on the shift.
3. We have also considered each location can have multiple Pubs because it not particular mentioned in the assessment.
4. A customer doesn't have to register to purchase goods. Customer is linked to transaction over order, this is to make comparison between the number of goods purchased between members simpler each item of stock has only one retailer

where it is purchased Order displays the total transactions of a single visit/multiple items purchased. Stock will be different in each pub.

5. We also considered that pubs can give discounts. For this we have created the Discounts table which is linked to menu item through item_id foreign key in discounts table.
6. Phone numbers for the customers are not mandatory as some customers don not want to share their phone number as per their privacy.

Task T3: Data Dictionary/Scripts

Data Dictionary

1) Table Name: Stocks

Stocks					
Attribute name	PK or AK	Data Type & Size	Domain and constraints	FK references	Description (Where non-obvious)
stock_item_id	PK	Int	Not null, AUTO_INCREMENT, UNIQUE		This field is also an auto increment (AUTO_INCREMENT)
pub_id		Int	Not null	Pubs	
item_name		varchar (50)	Not null		
quantity		BIGINT			Can be null because item may get out of stock = 0
expiery_date		DATETIME	Not null		
Import_date		DATETIME	Not null		
scale		varchar(30)	Not null		

2) Table Name: Location

Locations					
Attribute name	PK or AK	Data Type & Size	Domain and constraints	FK references	Description (Where non-obvious)
location_id	PK	Int	Not null, AUTO_INCREMENT, UNIQUE		This field is also an auto increment (AUTO_INCREMENT)

location_name		varchar(50)	Not null		
coordinates		varchar(30)	Can Be Null		Coordinates will tell the exact location of the city or state

3) Table Name: Pubs

Pubs					
Attribute name	PK or AK	Data Type & Size	Domain and constraints	FK references	Description (Where non-obvious)
pub_id	PK	Int	Not null, AUTO_INCREMENT, UNIQUE		This field is also an auto increment (AUTO_INCREMENT)
pub_name		varchar(30)	Not null		
location_id		Int	Not null	Locations	This field will tell where the location is situated
pub_telephone		varchar(15)	Not Null		E.164 states only 15 characters
address		varchar(30)			This can be a composite attribute
postcode		varchar(30)			This can be a composite attribute

4) Table Name: Menus

Menus					
Attribute name	PK or AK	Data Type & Size	Domain and constraints	FK references	Description (Where non-obvious)
item_id	PK	Int	Not null, AUTO_INCREMENT, UNIQUE		This field is also an auto increment (AUTO_INCREMENT)
pub_id		int	Not null	Pubs	
item_name		varchar (30)	Not null		
price		float	Not null		
menu_type		varchar (20)	Not null		Menu Item can be two types DRINK OR FOOD

5) Table Name: Orders

Orders					
Attribute name	PK or AK	Data Type & Size	Domain and constraints	FK references	Description (Where non-obvious)
order_id	P K	Int	Not null, AUTO_INCREMENT, UNIQUE		This field is also an auto increment (AUTO_INCREMENT)
transaction_id		Int	Not null	Transactions	
order_total		Int	Not null		
order_paid		BOOLEAN	Not null		
order_date_time		DATETIME	Not null		
payment_method		varchar(20)	Not null		This field can be cash or card

6) Tables Name: Transactions

Transactions					
Attribute name	PK or AK	Data Type & Size	Domain and constraints	FK references	Description (Where non-obvious)
transaction_id	P K	Int	Not null, AUTO_INCREMENT, UNIQUE		This field is an auto increment (AUTO_INCREMENT)
pub_id		Int	Not null	Pubs	
item_id		Int	Not null	Menus	
customer_id		Int	Not null	Customers	
transaction_date_time		DATETIME	Not null		CURRENT TIMESTAMP when the transaction success happened

7) Table Name: Customers

Customers					
Attribute name	PK or AK	Data Type & Size	Domain and constraints	FK references	Description (Where non-obvious)

customer_id	PK	Int	Not null, AUTO_INCREMENT, UNIQUE		This field is also an auto increment (AUTO_INCREMENT)
name		varchar (40)	Not null		
contact_number	AK	varchar(15)	Can be null		Customer doesn't need to provide number due to privacy
address		varchar (60)	Can be null		This field can be null
email	AK	varchar (50)	Not null		This field can be null
pub_id		Int	Not null	Pubs	

8) Table Name: Employees

Employees					
Attribute name	PK or AK	Data Type & Size	Domain and constraints	FK references	Description (Where non-obvious)
employee_id	PK	Int	Not null, AUTO_INCREMENT, UNIQUE		This field is also an auto increment (AUTO_INCREMENT)
role		varchar (30)	Not null		
name		varchar (50)	Not null		
contract_type		varchar (30)	Not null		
pub_id		Int	Not null		
address		varchar(100)	Not null		
contact_number		varchar(15)	Not null		E.164 states only 15 characters
postcode		varchar(20)	Not null		
email	AK	varchar(100)	Not null		

9) Table Name: Employees shifts

Employee Shifts					
Attribute name	PK or AK	Data Type & Size	Domain and constraints	FK references	Description (Where non-obvious)

employee_shift_id	P K	Int	Not null, AUTO_INCREMENT, UNIQUE		This field is also an auto increment (AUTO_INCREMENT)
employee_id		Int	Not null	Employee Table	
shift_start		DATETIME	Not null		This column will record employee shift start time and date
shift_end		DATETIME	Not null		This column will record employee shift end time and date
on_shift		Boolean	Not Null		This value can be true or false 1 (true) or 0 (false).

10) Table Name: Discounts

Discounts (This table can be null if there are no discounts but if there is a record each column should exist for the record)					
Attribute name	PK or AK	Data Type & Size	Domain and constraints	FK references	Description (Where non-obvious)
discount_id	P K	Int	Not null, AUTO_INCREMENT, UNIQUE		This field is also an auto increment (AUTO_INCREMENT)
item_id		Int	Not null	Item Table	
discount_percentage		Int	Not null		
end_date		Date	Not null		
previously_used		Boolean	Not null		This value can be true or false 1 (true) or 0 (false).

Before querying the data from the database, we have used following queries to create the database table and for inserting the fake data these queries can be found in the APPENDIX.

Scripts To Create Tables

For creating the table's used queries are following.

1 Table Name: Locations

```
CREATE TABLE Locations (  
  location_id      int NOT NULL AUTO_INCREMENT UNIQUE,  
  location_name    varchar(50) NOT NULL,  
  coordinates      varchar(20),  
  
  PRIMARY KEY (location_id)  
  
);
```

2 Table Name: Pubs

```
CREATE TABLE Pubs (  
  pub_id          int NOT NULL AUTO_INCREMENT UNIQUE,  
  pub_name        varchar(50) NOT NULL,  
  pub_telephone   varchar(15) NOT NULL,  
  location_id     int NOT NULL,  
  address         varchar(30),  
  pincode         varchar(20),  
  
  PRIMARY KEY (pub_id),  
  
  CONSTRAINT FK_PubsLocation FOREIGN KEY (location_id)  
  REFERENCES Locations (location_id)  
  
);
```

3 Table Name: Menus

```
CREATE TABLE Menus (  
  
  item_id int NOT NULL AUTO_INCREMENT UNIQUE,  
  menu_type varchar(20) NOT NULL,  
  item_name varchar(30) NOT NULL,  
  pub_id int NOT NULL,  
  price float NOT NULL,  
  
  PRIMARY KEY (item_id),  
  
  CONSTRAINT FK_MenuVenus FOREIGN KEY (pub_id)  
  REFERENCES Pubs (pub_id)  
  
);
```

4 Table Name: Customers

```
CREATE TABLE Customers (  
  customer_id int NOT NULL AUTO_INCREMENT UNIQUE,  
  pub_id int,  
  name varchar(30) NOT NULL,  
  contact_number varchar(15),  
  address varchar(60),
```

```

email      varchar(50) NOT NULL,

PRIMARY KEY (customer_id),

CONSTRAINT FK_CustomerPub FOREIGN KEY (pub_id)
REFERENCES Pubs (pub_id)

);

```

5 Table Name: Transactions

```

CREATE TABLE Transactions (
    transaction_id int NOT NULL AUTO_INCREMENT UNIQUE,
    pub_id int NOT NULL,
    item_id int NOT NULL,
    customer_id int NOT NULL,
    transaction_date_time DATETIME NOT NULL,

    PRIMARY KEY (transaction_id),

    CONSTRAINT FK_TransactionPub FOREIGN KEY (pub_id)
    REFERENCES Pubs (pub_id),

    CONSTRAINT FK_TransactionMenuItem FOREIGN KEY (item_id)
    REFERENCES Menus (item_id),

    CONSTRAINT FK_TransactionCustomer FOREIGN KEY (customer_id)
    REFERENCES Customers (customer_id)

);

```

6 Table Name: Orders

```

CREATE TABLE Orders (
    order_id int NOT NULL AUTO_INCREMENT UNIQUE,
    transaction_id int NOT NULL,
    order_total int NOT NULL,
    order_paid BOOLEAN NOT NULL,
    order_date date NOT NULL,
    payment_method varchar(20) NOT NULL,

    PRIMARY KEY (order_id),

    CONSTRAINT FK_OrderTransaction FOREIGN KEY (transaction_id)
    REFERENCES Transactions (transaction_id)

);

```

7 Table Name: Employees

```

CREATE TABLE Employees (

    employee_id int NOT NULL AUTO_INCREMENT UNIQUE,
    pub_id int,
    role varchar(20) NOT NULL,
    name varchar(50) NOT NULL,

```

```

contact_number varchar(15) NOT NULL,
contract_type varchar(30) NOT NULL,
address varchar(100) NOT NULL,
postcode varchar(20) NOT NULL,

PRIMARY KEY (employee_id),
CONSTRAINT FK_EmployeePub FOREIGN KEY (pub_id)
REFERENCES Pubs (pub_id)

);

```

8 Table Name: Employeeshifts

```

CREATE TABLE Employeeshifts (
    employee_shift_id int NOT NULL AUTO_INCREMENT UNIQUE,
    employee_id int NOT NULL,
    shift_start DATETIME NOT NULL,
    shift_end DATETIME NOT NULL,
    on_shift BOOLEAN NOT NULL,

    PRIMARY KEY (employee_shift_id),
    CONSTRAINT FK_EmployeeEmployeeShift FOREIGN KEY (employee_id)
    REFERENCES Employees (employee_id)

);

```

9 Table Name: Stocks

```

CREATE TABLE Stocks (
    stock_id int NOT NULL AUTO_INCREMENT UNIQUE,
    pub_id int NOT NULL,
    item_name varchar(50) NOT NULL,
    quantity BIGINT NOT NULL,

    expiry_date DATETIME NOT NULL,
    import_date DATETIME NOT NULL,

    PRIMARY KEY (stock_id),
    CONSTRAINT FK_StockPub FOREIGN KEY (pub_id)
    REFERENCES Pubs (pub_id)

);

```

10 Table Name: Discounts

```

CREATE TABLE Discounts (
    discount_id int NOT NULL AUTO_INCREMENT UNIQUE,
    item_id int NOT NULL,
    discount_percentage int NOT NULL,
    end_date DATETIME NOT NULL,
    previously_used BOOLEAN NOT NULL,

    PRIMARY KEY (discount_id),
    CONSTRAINT FK_DiscountItem FOREIGN KEY (item_id)
    REFERENCES Menus (item_id)

);

```

For Altering the table, we have used following statement

For Adding New Column to the customers
ALTER TABLE Customers
ADD COLUMN contact_number varchar(15);

Similarly using alter we can alter any column in particular table.

Task T4: SQL Queries/ Final Demonstration

SQL Queries

FIVE SQL statements that reflect the needs of the business printout of the tables, data, and the queries you are running.

Five Query Representing the Business Logic

Query one:

```
SELECT
  Customers.name AS Customer_name,
  Customers.email,
  Pubs.pub_name,
  Locations.location_name,
  Locations.coordinates
FROM Customers
JOIN Pubs
  ON Customers.pub_id = Pubs.pub_id
JOIN Locations
  ON Pubs.location_id = Locations.location_id
ORDER BY customer_id DESC;
```

Description: Following query will select customer's name, email, pub_name, location name, location coordinates from three different tables and this query can be used by restaurant managers to know which pub is more crowded and details of the customer who visited.

Query two:

```
SELECT Pubs.pub_name, Transactions.transaction_date_time,
SUM(order_total) AS Total_Sales,
Transactions.transaction_id
FROM Orders
INNER JOIN Transactions
```

```
ON Orders.transaction_id = Transactions.transaction_id
JOIN Pubs
ON Pubs.pub_id = Transactions.pub_id
GROUP BY Transactions.transaction_id;
```

Description: This query will select transaction happened on particular pub with date and time including total sales on that particular date this can be used by owners of restaurant keep the track of the of the sales on daily bases according to the pub.

Query three:

```
SELECT Pubs.pub_id, Pubs.pub_name AS Resturant_name,
Pubs.address,
COUNT(employee_id) AS Number_of_employees
FROM Employees
JOIN Pubs
ON Pubs.pub_id = Employees.pub_id
GROUP BY Pubs.pub_id;
```

Description: Total Number of employees according to the pub and there address this query can be used by management to look at the manpower in each pub and recruit accordingly.

Query four:

```
SELECT Employees.employee_id,
Employees.name,
Employeeshifts.shift_start,
Employeeshifts.shift_end
From Employeeshifts
INNER JOIN Employees
ON Employees.employee_id = Employeeshifts.employee_id
WHERE Employeeshifts.on_shift = true
ORDER BY employee_id ASC;
```

Description: This query will fetch employee working details and shows if the employee is on shift and their timings this query can be used by managers to look at restaurant staff shifts and will show which employee is currently working.

Query Five:

```
SELECT CONCAT(Menus.item_name, ' - ', Menus.price),
CONCAT(Pubs.pub_id, Pubs.pub_name),
Pubs.address
FROM Menus
JOIN Pubs
ON Pubs.pub_id = Menus.pub_id
ORDER BY item_name;
```

Description: This will show which menu belongs to which pub (Eg: Which restaurant have which menu and their items) this can be used by owners of the restaurant to decide on what price to put on items and which are already existing.

Query SIX:

```
SELECT
Customers.name,
Customers.email,
Transactions.transaction_date_time,
Menus.item_name,
Orders.order_total, Orders.payment_method
FROM Customers
JOIN Transactions
    ON Customers.customer_id = Transactions.customer_id
JOIN Menus
    ON Menus.item_id = Transactions.item_id
JOIN Orders
    ON Orders.transaction_id = Transactions.transaction_id
ORDER BY transaction_date_time DESC;
```

Description: This query will fetch customer name, email, transactions, order item, order total, and payment method from four different tables this can be used to know which customer have paid how much and payment method used by sales team to look at which item has customer have bought its cost with payment method used to pay.

Demonstration

Query 1 Screenshot

```
mysql> SELECT
-> Customers.name AS Customer_name,
-> Customers.email,
-> Pubs.pub_name,
-> Locations.location_name,
-> Locations.coordinates
-> FROM Customers
-> JOIN Pubs
->   ON Customers.pub_id = Pubs.pub_id
-> JOIN Locations
->   ON Pubs.location_id = Locations.location_id
-> ORDER BY customer_id DESC;
```

Customer_name	email	pub_name	location_name	coordinates
Habergham	thabergham2p@seesaa.net	Eryngium constancei Sheikh	Mizi	112.520954
Benezet	dbenezet2o@cam.ac.uk	Juncus validus Coville	Baltasar Brum	-57.3298733
Gwatkin	ggwatkin2n@dedecms.com	Drosera intermedia Hayne	Bani	119.8572781
Abbot	mabbot2m@skyrock.com	Bromus racemosus L.	Hauhena	123.9182
Trevino	wtrevino2l@usgs.gov	Megalaria grossa (Pers. ex Nyl.) Hafellner	Rušanj	20.4439853
Magner	dmagner2j@paypal.com	Buglossoides arvensis (L.) I.M. Johnst.	Panguang	121.3729026
Cottrill	lcottrill2g@aboutads.info	Draba helleriana Greene var. blumeri C.L. Hitchc.	New Yekepa	-8.5363982
Twaits	mtwaits2f@mashable.com	Pedicularis dudleyi Elmer	Ryazanskiy	37.7808815
Catchpole	bcatchpole2e@imageshack.us	Rhizocarpon lindsayanum Rasanen	Same	125.6478189
Hyde-Chambers	phydechambers2d@discuz.net	Rajania cordata L.	Várzea	-8.5747445
Hundley	chundley2c@barnesandnoble.com	Lycopus rubellus Moench	Pinhal Novo	-8.9095861
Lanfear	blanfear2b@youtube.com	Cryptantha barnebyi I.M. Johnst.	Dazhongzhuang	114.387043

Query 2 Screenshot

```
mysql> SELECT Pubs.pub_name, Transactions.transaction_date_time,
-> SUM(order_total) AS Total_Sales,
-> Transactions.transaction_id
-> FROM Orders
-> INNER JOIN Transactions
-> ON Orders.transaction_id = Transactions.transaction_id
-> JOIN Pubs
-> ON Pubs.pub_id = Transactions.pub_id
-> GROUP BY Transactions.transaction_id;
```

pub_name	transaction_date_time	Total_Sales	transaction_id
NULL	2021-12-18 02:36:54	200	1
Cuscuta pentagona Engelm. var. pentagona	2021-04-07 00:00:00	72	3
Silene petersonii Maguire var. petersonii	2021-01-28 00:00:00	91	4
Yucca treculeana Carrière	2021-03-09 00:00:00	88	5
Melica porteri Scribn. var. laxa Boyle	2021-12-21 00:00:00	14	6
Potamogeton xnitens Weber (pro sp.)	2021-02-16 00:00:00	85	7
Digitaria abyssinica (Hochst. ex A. Rich.) Stapf	2021-02-07 00:00:00	74	8
Scrophularia oregana Pennell	2021-03-29 00:00:00	35	9
Cryptantha insolita (J.F. Macbr.) Payson	2021-02-09 00:00:00	88	10
Lycopodiella alopecuroides (L.) Cranfill	2021-03-14 00:00:00	28	11

Query 3 Screenshot

```
mysql> SELECT Pubs.pub_id, Pubs.pub_name AS Resturant_name,
-> Pubs.address,
-> COUNT(employee_id) AS Number_of_employees
-> FROM Employees
-> JOIN Pubs
-> ON Pubs.pub_id = Employees.pub_id
-> GROUP BY Pubs.pub_id;
```

pub_id	Resturant_name	address	Number_of_employees
1	NULL	Block 2 Fratton Road	1
3	Cuscuta pentagona Engelm. var. pentagona	2 Butterfield Street	1
4	Silene petersonii Maguire var. petersonii	68 David Hill	1
5	Yucca treculeana Carrière	9 Mariners Cove Court	1
6	Melica porteri Scribn. var. laxa Boyle	93755 Sheridan Point	1
8	Digitaria abyssinica (Hochst. ex A. Rich.) Stapf	4 Welch Lane	1
9	Scrophularia oregana Pennell	15608 Iowa Avenue	1
10	Cryptantha insolita (J.F. Macbr.) Payson	468 Norway Maple Parkway	1
11	Lycopodiella alopecuroides (L.) Cranfill	5 Sachtjen Center	1
13	Pinus discolor D.K. Bailey & Hawksw.	3025 Lien Parkway	1
14	Physocarpus opulifolius (L.) Maxim., orth. cons.	07024 Killdeer Pass	1
15	Arthonia pyrrhula Nyl.	1 Ridge Oak Alley	1

Query 4 Screenshot

```
mysql> SELECT CONCAT(Menus.item_name, '-', Menus.price),
-> CONCAT(Pubs.pub_id, Pubs.pub_name),
-> Pubs.address
-> FROM Menus
-> JOIN Pubs
-> ON Pubs.pub_id = Menus.pub_id
-> ORDER BY item_name;
```

CONCAT(Menus.item_name, '-', Menus.price)	CONCAT(Pubs.pub_id, Pubs.pub_name)	address
Amarula Cream-11	88Pedicularis dudleyi Elmer	31 Marquette Parkway
Bagel - Everything Presliced-14	51Coreopsis verticillata L.	9 Jana Place
Beef Dry Aged Tenderloin Aaa-23	15Arthonia pyrrhula Nyl.	1 Ridge Oak Alley
Beef Tenderloin Aaa-25	22Pertusaria subpertusa Brodo	8 Lindbergh Point
Beer-7.2	NULL	Block 2 Fratton Road
Bread - Roll, Canadian Dinner-3	6Melica porteri Scribn. var. laxa Boyle	93755 Sheridan Point
Butter - Pod-10	84Cryptantha barnebyi I.M. Johnst.	211 Stoughton Terrace
Cake Sheet Combo Party Pack-6	73Deschampsia setacea (Huds.) Hack.	1 Sommers Center
Cheese - Cheddar, Old White-6	95Bromus racemosus L.	7989 Elmside Point
Cheese - Parmesan Grated-23	50Umbilicaria polyrhiza (L.) Fr.	9 Lighthouse Bay Junction
Cheese - Sheep Milk-8	28Packera contermina (Greenm.) J.F. Bain	29008 Kedzie Place
Cheese - Swiss Sliced-16	19Galactia floridana Torr. & A. Gray	17 Hanover Crossing
Chef Hat 25cm-24	45Harpagonella A. Gray	80023 Debs Court
Chips - Doritos-15	25Cerastium gracile Dufour	2748 Bashford Street
Cocoa Powder - Natural-8	45Silene petersonii Maguire var. petersonii	68 David Hill
Coffee - 10oz Cup 92961-20	94Megalaria grossa (Pers. ex Nyl.) Hafellner	523 Claremont Point
Coffee - Almond Amaretto-27	9Scrophularia oregana Pennell	15608 Iowa Avenue

Query 5 Screenshot

```
mysql> SELECT CONCAT(Menus.item_name, '-', Menus.price),
-> CONCAT(Pubs.pub_id, Pubs.pub_name),
-> Pubs.address
-> FROM Menus
-> JOIN Pubs
-> ON Pubs.pub_id = Menus.pub_id
-> ORDER BY item_name;
```

CONCAT(Menus.item_name, '-', Menus.price)	CONCAT(Pubs.pub_id, Pubs.pub_name)	address
Amarula Cream-11	88Pedicularis dudleyi Elmer	31 Marquette Parkway
Bagel - Everything Presliced-14	51Coreopsis verticillata L.	9 Jana Place
Beef Dry Aged Tenderloin Aaa-23	15Arthonia pyrrhula Nyl.	1 Ridge Oak Alley
Beef Tenderloin Aaa-25	22Pertusaria subpertusa Brodo	8 Lindbergh Point
Beer-7.2	NULL	Block 2 Fratton Road
Bread - Roll, Canadian Dinner-3	6Melica porteri Scribn. var. laxa Boyle	93755 Sheridan Point
Butter - Pod-10	84Cryptantha barnebyi I.M. Johnst.	211 Stoughton Terrace
Cake Sheet Combo Party Pack-6	73Deschampsia setacea (Huds.) Hack.	1 Sommers Center
Cheese - Cheddar, Old White-6	95Bromus racemosus L.	7989 Elside Point
Cheese - Parmesan Grated-23	50Umbilicaria polyrhiza (L.) Fr.	9 Lighthouse Bay Junction
Cheese - Sheep Milk-8	28Packera contermina (Greenm.) J.F. Bain	29008 Kedzie Place
Cheese - Swiss Sliced-16	19Galactia floridana Torr. & A. Gray	17 Hanover Crossing
Chef Hat 25cm-24	45Harpagonella A. Gray	80023 Debs Court
Chips - Doritos-15	25Ceratium gracile Dufour	2748 Bashford Street
Cocoa Powder - Natural-8	4Silene petersonii Maguire var. petersonii	68 David Hill
Coffee - 10oz Cup 92961-20	94Megalania grossa (Pers. ex Nyl.) Hafellner	523 Claremont Point
Coffee - Almond Amaretto-27	9Scrophularia oregana Pennell	15608 Iowa Avenue

Query 6 Screenshot

```
mysql> SELECT
-> Customers.name,
-> Customers.email,
-> Transactions.transaction_date_time,
-> Menus.item_name,
-> Orders.order_total, Orders.payment_method
-> FROM Customers
-> JOIN Transactions
-> ON Customers.customer_id = Transactions.customer_id
-> JOIN Menus
-> ON Menus.item_id = Transactions.item_id
-> JOIN Orders
-> ON Orders.transaction_id = Transactions.transaction_id
-> ORDER BY transaction_date_time DESC;
```

name	email	transaction_date_time	item_name	order_total	payment_method
Wicken	kwicken25@tumblr.com	2021-12-25 00:00:00	Seaweed Green Sheets	61	NULL
Catchpole	bcatchpole2e@imageshack.us	2021-12-24 00:00:00	Ice Cream - Super Sandwich	100	NULL
Abbot	mabbot2m@skyrock.com	2021-12-22 00:00:00	Cheese - Cheddar, Old White	7	NULL
Kilrow	ekilrow5@netscape.com	2021-12-21 00:00:00	Bread - Roll, Canadian Dinner	14	NULL
Rayapudi	NULL	2021-12-18 02:36:54	Fish & Chip	100	NULL
Rayapudi	NULL	2021-12-18 02:36:54	Fish & Chip	100	NULL
Tabary	mtabary1y@newsvine.com	2021-12-16 00:00:00	Nori Sea Weed - Gold Label	2	NULL
Clears	dclearsj@lycos.com	2021-12-16 00:00:00	Sour Cream	19	NULL
Bertelsen	sbertelsen@marriott.com	2021-12-13 00:00:00	Green Tea Refresher	22	NULL
Berrisford	gberrisford1j@mozilla.org	2021-12-04 00:00:00	Flour - Rye	89	NULL

Note: More depth demonstration will be provided in live demonstration

References

SQL Tutorial. (n.d.). W3Schools. <https://www.w3schools.com/sql/>

D. (2021, November 19). How to Get the Data Type of Columns in SQL Server. Data to Fish. <https://datatofish.com/data-type-columns-sql-server/>

Matthew, O. (2021, October 25). Gathering Requirements [Slides]. Gathering Requirements. <https://moodle.port.ac.uk>

Matthew, O. (2021b, December 29). Database Management Lecture 4 [Slides]. University of Portsmouth. <https://moodle.port.ac.uk>

MySQL:: MySQL Documentation. (n.d.). MySql. Retrieved 15 December 2021, from <https://dev.mysql.com/doc/>

Matthew, O.(2021) Database Design Data Models Entity Relationship Approach. Database Management – Lecture 3

Matthew, O. (2022d, January 3). Normalisation Lecture 5 [Slides]. University of Portsmouth. <https://moodle.port.ac.uk>

Appendix

Following data was inserted in respective tables and these data was generated using <https://mockaroo.com>.

1	<pre>insert into Locations (location_id, location_name, coordinates) values (1, 'Bayzhansay', 69.9222782); insert into Locations (location_id, location_name, coordinates) values (2, 'Al Maḥallah al Kubrá', 31.162448); insert into Locations (location_id, location_name, coordinates) values (3, 'Qingquan', 112.0069982); insert into Locations (location_id, location_name, coordinates) values (4, 'Ivot', 104.292306); insert into Locations (location_id, location_name, coordinates) values (5, 'Tumauini', 121.834345); insert into Locations (location_id, location_name, coordinates) values (6, 'Zahlé', 35.9019775); insert into Locations (location_id, location_name, coordinates) values (7, 'Lanhe', 113.343022); insert into Locations (location_id, location_name, coordinates) values (8, 'Vyškov', 17.0367923); insert into Locations (location_id, location_name, coordinates) values (9, 'Hamilton', -64.7858887); insert into Locations (location_id, location_name, coordinates) values (10, 'Sampaio', -9.0952451);</pre>
2	<pre>insert into Pubs (pub_id, pub_name, pub_telephone, location_id, address, pincode) values (3, 'Cuscuta pentagona Engelm. var. pentagona', '120-697-3055', 3, '2 Butterfield Street', null); insert into Pubs (pub_id, pub_name, pub_telephone, location_id, address, pincode) values (4, 'Silene petersonii Maguire var. petersonii', '510-913-7894', 4, '68 David Hill', null); insert into Pubs (pub_id, pub_name, pub_telephone, location_id, address, pincode) values (5, 'Yucca treculeana Carrière', '882-904-4518', 5, '9 Mariners Cove Court', '2253'); insert into Pubs (pub_id, pub_name, pub_telephone, location_id, address, pincode) values (6, 'Melica porteri Scribn. var. laxa Boyle', '315-485-4396', 6, '93755 Sheridan Point', '9604'); insert into Pubs (pub_id, pub_name, pub_telephone, location_id, address, pincode) values (7, 'Potamogeton ×nitens Weber (pro sp.)', '235-145-3856', 7, '44220 Spohn Trail', '301420'); insert into Pubs (pub_id, pub_name, pub_telephone, location_id, address, pincode) values (8, 'Digitaria abyssinica (Hochst. ex A. Rich.) Stapf', '764-255-5829', 8, '4 Welch Lane', '912 24'); insert into Pubs (pub_id, pub_name, pub_telephone, location_id, address, pincode) values (9, 'Scrophularia oregana Pennell', '704-921-3975', 9, '15608 Iowa Avenue', null);</pre>

	<p>insert into Pubs (pub_id, pub_name, pub_telephone, location_id, address, pincode) values (10, 'Cryptantha insolita (J.F. Macbr.) Payson', '397-992-3896', 10, '468 Norway Maple Parkway', null);</p> <p>insert into Pubs (pub_id, pub_name, pub_telephone, location_id, address, pincode) values (11, 'Lycopodiella alopecuroides (L.) Cranfill', '945-361-9679', 11, '5 Sachtjen Center', null);</p>
3	<p>insert into Menus (item_id, menu_type, item_name, pub_id, price) values (1, null, 'Eggs - Extra Large', 1, '24.13');</p> <p>insert into Menus (item_id, menu_type, item_name, pub_id, price) values (2, null, 'Chips Potato Salt Vinegar 43g', 2, '7.89');</p> <p>insert into Menus (item_id, menu_type, item_name, pub_id, price) values (3, null, 'Jam - Blackberry, 20 Ml Jar', 3, '19.27');</p> <p>insert into Menus (item_id, menu_type, item_name, pub_id, price) values (4, null, 'Cake - Mini Potato Pancake', 4, '20.51');</p> <p>insert into Menus (item_id, menu_type, item_name, pub_id, price) values (5, null, 'Pecan Raisin - Tarts', 5, '14.15');</p> <p>insert into Menus (item_id, menu_type, item_name, pub_id, price) values (6, null, 'Pineapple - Regular', 6, '29.83');</p> <p>insert into Menus (item_id, menu_type, item_name, pub_id, price) values (7, null, 'Cabbage - Savoy', 7, '4.66');</p> <p>insert into Menus (item_id, menu_type, item_name, pub_id, price) values (8, null, 'Lemonade - Kiwi, 591 Ml', 8, '27.09');</p> <p>insert into Menus (item_id, menu_type, item_name, pub_id, price) values (9, null, 'Alize Red Passion', 9, '7.32');</p> <p>insert into Menus (item_id, menu_type, item_name, pub_id, price) values (10, null, 'Lettuce - Boston Bib', 10, '2.15');</p> <p>insert into Menus (item_id, menu_type, item_name, pub_id, price) values (11, null, 'Table Cloth 54x54 Colour', 11, '16.45');</p>
4	<p>Insert into customers values ('1', '1', 'Rayapudi', '7341206839');</p>
5	<p>Insert into transactions values (`1`, `1`, `1`, ` 2021-12-18 02:36:54.480`, `card`);</p> <p>insert into Transactions (transaction_id, pub_id, item_id, customer_id, transaction_date_time) values (1, 1, 1, 1, '2021-06-04');</p> <p>insert into Transactions (transaction_id, pub_id, item_id, customer_id, transaction_date_time) values (2, 2, 2, 2, '2021-11-18');</p> <p>insert into Transactions (transaction_id, pub_id, item_id, customer_id, transaction_date_time) values (3, 3, 3, 3, '2021-06-05');</p> <p>insert into Transactions (transaction_id, pub_id, item_id, customer_id, transaction_date_time) values (4, 4, 4, 4, '2021-07-04');</p> <p>insert into Transactions (transaction_id, pub_id, item_id, customer_id, transaction_date_time) values (5, 5, 5, 5, '2021-12-01');</p>

	<pre> insert into Transactions (transaction_id, pub_id, item_id, customer_id, transaction_date_time) values (6, 6, 6, 6, '2021-10- 28'); insert into Transactions (transaction_id, pub_id, item_id, customer_id, transaction_date_time) values (7, 7, 7, 7, '2021-02- 27'); insert into Transactions (transaction_id, pub_id, item_id, customer_id, transaction_date_time) values (8, 8, 8, 8, '2021-09- 23'); insert into Transactions (transaction_id, pub_id, item_id, customer_id, transaction_date_time) values (9, 9, 9, 9, '2021-06- 29'); insert into Transactions (transaction_id, pub_id, item_id, customer_id, transaction_date_time) values (10, 10, 10, 10, '2021- 09-29'); insert into Transactions (transaction_id, pub_id, item_id, customer_id, transaction_date_time) values (14, 14, 14, 14, '2021- 12-03'); </pre>
6	<pre> Insert into orders values ('1', '1', '100£', '1', '2021-12-18'); insert into Orders (order_id, transaction_id, order_total, order_paid, order_date, payment_method) values (1, 1, 30, false, '2021-11-11', null); insert into Orders (order_id, transaction_id, order_total, order_paid, order_date, payment_method) values (2, 2, 14, false, '2021-04-25', null); insert into Orders (order_id, transaction_id, order_total, order_paid, order_date, payment_method) values (3, 3, 41, true, '2021-01-09', null); insert into Orders (order_id, transaction_id, order_total, order_paid, order_date, payment_method) values (4, 4, 15, true, '2021-03-11', null); insert into Orders (order_id, transaction_id, order_total, order_paid, order_date, payment_method) values (5, 5, 60, false, '2021-09-01', null); insert into Orders (order_id, transaction_id, order_total, order_paid, order_date, payment_method) values (6, 6, 92, false, '2021-08-13', null); insert into Orders (order_id, transaction_id, order_total, order_paid, order_date, payment_method) values (7, 7, 94, true, '2021-05-25', null); insert into Orders (order_id, transaction_id, order_total, order_paid, order_date, payment_method) values (8, 8, 14, false, '2021-08-09', null); insert into Orders (order_id, transaction_id, order_total, order_paid, order_date, payment_method) values (9, 9, 27, false, '2021-03-03', null); insert into Orders (order_id, transaction_id, order_total, order_paid, order_date, payment_method) values (10, 10, 66, true, '2021-10-05', null); </pre>
7	<pre> Insert into employees values ('1', '1', 'Manager', 'Alex', '7341206839', 'Full-time', 'Portsmouth United Kingdom PO152DJ'); insert into Employees (employee_id, pub_id, role, name, contact_number, address, postcode, contract_type) values (1, 1, 'Construction Expeditor', 'Shreeve', '389-606- 1320', '76 Arrowood Way', null, 'Social Worker'); insert into Employees (employee_id, pub_id, role, name, contact_number, address, postcode, contract_type) values (2, 2, 'Construction Expeditor', 'Honig', '786-737-9078', '65376 Sherman Point', null, 'Pharmacist'); insert into Employees (employee_id, pub_id, role, name, contact_number, address, postcode, contract_type) values (3, 3, 'Subcontractor', 'Mawer', '179-974-3842', '9303 Corscot Court', '681 22', 'Cost Accountant'); insert into </pre>

	<p>Employees (employee_id, pub_id, role, name, contact_number, address, postcode, contract_type) values (4, 4, 'Construction Manager', 'Andriveaux', '305-816-0546', '299 Almo Terrace', '1709', 'Administrative Assistant II'); insert into Employees (employee_id, pub_id, role, name, contact_number, address, postcode, contract_type) values (5, 5, 'Engineer', 'Fosh', '356-202-7907', '25890 Eagan Pass', '361606', 'VP Sales'); insert into Employees (employee_id, pub_id, role, name, contact_number, address, postcode, contract_type) values (6, 6, 'Engineer', 'Rosekilly', '402-600-1974', '9442 Gulseth Point', '70170', 'Associate Professor'); insert into Employees (employee_id, pub_id, role, name, contact_number, address, postcode, contract_type) values (7, 7, 'Construction Expeditor', 'Tinniswood', '974-244-4531', '7 Novick Way', null, 'Geologist I'); insert into Employees (employee_id, pub_id, role, name, contact_number, address, postcode, contract_type) values (8, 8, 'Construction Worker', 'O"Scollee', '312-168-3249', '7 Briar Crest Lane', '59407 CEDEX', 'Operator'); insert into Employees (employee_id, pub_id, role, name, contact_number, address, postcode, contract_type) values (9, 9, 'Construction Worker', 'Parminter', '744-846-7801', '2618 South Place', '4705-029', 'Senior Developer'); insert into Employees (employee_id, pub_id, role, name, contact_number, address, postcode, contract_type) values (10, 10, 'Architect', 'Mytton', '652-596-0640', '7 West Plaza', null, 'Food Chemist');</p>
8	<p>insert into Employeeeshifts (employee_shift_id, employee_id, shift_start, shift_end, on_shift) values (1, 1, '2021-08-21', '2021-08-14', true); insert into Employeeeshifts (employee_shift_id, employee_id, shift_start, shift_end, on_shift) values (2, 2, '2021-02-16', '2021-06-05', false); insert into Employeeeshifts (employee_shift_id, employee_id, shift_start, shift_end, on_shift) values (3, 3, '2021-05-06', '2021-11-24', true); insert into Employeeeshifts (employee_shift_id, employee_id, shift_start, shift_end, on_shift) values (4, 4, '2021-09-07', '2021-01-14', false); insert into Employeeeshifts (employee_shift_id, employee_id, shift_start, shift_end, on_shift) values (5, 5, '2021-01-17', '2020-12-30', true); insert into Employeeeshifts (employee_shift_id, employee_id, shift_start, shift_end, on_shift) values (6, 6, '2021-10-28', '2021-08-17', false); insert into Employeeeshifts (employee_shift_id, employee_id, shift_start, shift_end, on_shift) values (7, 7, '2021-08-04', '2021-06-25', true); insert into Employeeeshifts (employee_shift_id, employee_id, shift_start, shift_end, on_shift) values (8, 8, '2021-06-23', '2021-10-08', false); insert into Employeeeshifts (employee_shift_id, employee_id, shift_start, shift_end, on_shift) values (9, 9, '2021-12-15', '2021-09-24', false); insert into Employeeeshifts (employee_shift_id, employee_id, shift_start, shift_end, on_shift) values (10, 10, '2021-12-15', '2021-11-23', false); insert into Employeeeshifts (employee_shift_id, employee_id, shift_start, shift_end, on_shift) values (11, 11, '2020-12-30', '2021-08-17', false); insert into Employeeeshifts (employee_shift_id, employee_id, shift_start, shift_end, on_shift) values (12, 12, '2021-01-07', '2021-10-28', false); insert into Employeeeshifts (employee_shift_id, employee_id, shift_start, shift_end, on_shift) values (13, 13, '2021-10-29', '2021-07-23', false); insert into Employeeeshifts (employee_shift_id, employee_id, shift_start, shift_end, on_shift) values (14, 14, '2021-10-11', '2021-05-08', true); insert into Employeeeshifts (employee_shift_id, employee_id, shift_start, shift_end, on_shift) values (15, 15, '2021-12-15', '2021-07-18', false); insert into Employeeeshifts (employee_shift_id, employee_id, shift_start, shift_end, on_shift) values (16, 16, '2021-09-26', '2020-12-31', true); insert into Employeeeshifts (employee_shift_id, employee_id, shift_start, shift_end, on_shift) values (17, 17, '2021-01-14', '2021-08-22', true); insert into Employeeeshifts (employee_shift_id, employee_id, shift_start, shift_end, on_shift) values (18, 18, '2021-11-18', '2021-08-23', true); insert into Employeeeshifts (employee_shift_id,</p>

	employee_id, shift_start, shift_end, on_shift) values (19, 19, '2021-06-28', '2021-08-11', false); insert into EmployeeShifts (employee_shift_id, employee_id, shift_start, shift_end, on_shift) values (20, 20, '2021-03-17', '2021-05-05', false); insert into EmployeeShifts (employee_shift_id, employee_id, shift_start, shift_end, on_shift) values (21, 21, '2021-11-06', '2021-09-05', true);
9	Insert into stocks values ('1', '1', 'Onions', '2022-12-18 02:36:54.480', '2021-12-18 02:36:54.480');
10	insert into Discounts (discount_id, item_id, discount_percentage, end_date, previously_used) values (1, 1, 85, '2021-04-11', false); insert into Discounts (discount_id, item_id, discount_percentage, end_date, previously_used) values (2, 2, 26, '2021-05-10', false); insert into Discounts (discount_id, item_id, discount_percentage, end_date, previously_used) values (3, 3, 22, '2021-09-04', true); insert into Discounts (discount_id, item_id, discount_percentage, end_date, previously_used) values (4, 4, 74, '2021-01-18', false); insert into Discounts (discount_id, item_id, discount_percentage, end_date, previously_used) values (5, 5, 5, '2021-05-28', false); insert into Discounts (discount_id, item_id, discount_percentage, end_date, previously_used) values (6, 6, 46, '2021-01-04', false); insert into Discounts (discount_id, item_id, discount_percentage, end_date, previously_used) values (7, 7, 60, '2021-05-22', true); insert into Discounts (discount_id, item_id, discount_percentage, end_date, previously_used) values (8, 8, 60, '2021-12-20', false); insert into Discounts (discount_id, item_id, discount_percentage, end_date, previously_used) values (9, 9, 25, '2021-07-07', false); insert into Discounts (discount_id, item_id, discount_percentage, end_date, previously_used) values (10, 10, 93, '2021-10-19', false);