Company Overview

Alejandro's
RESTAURANT

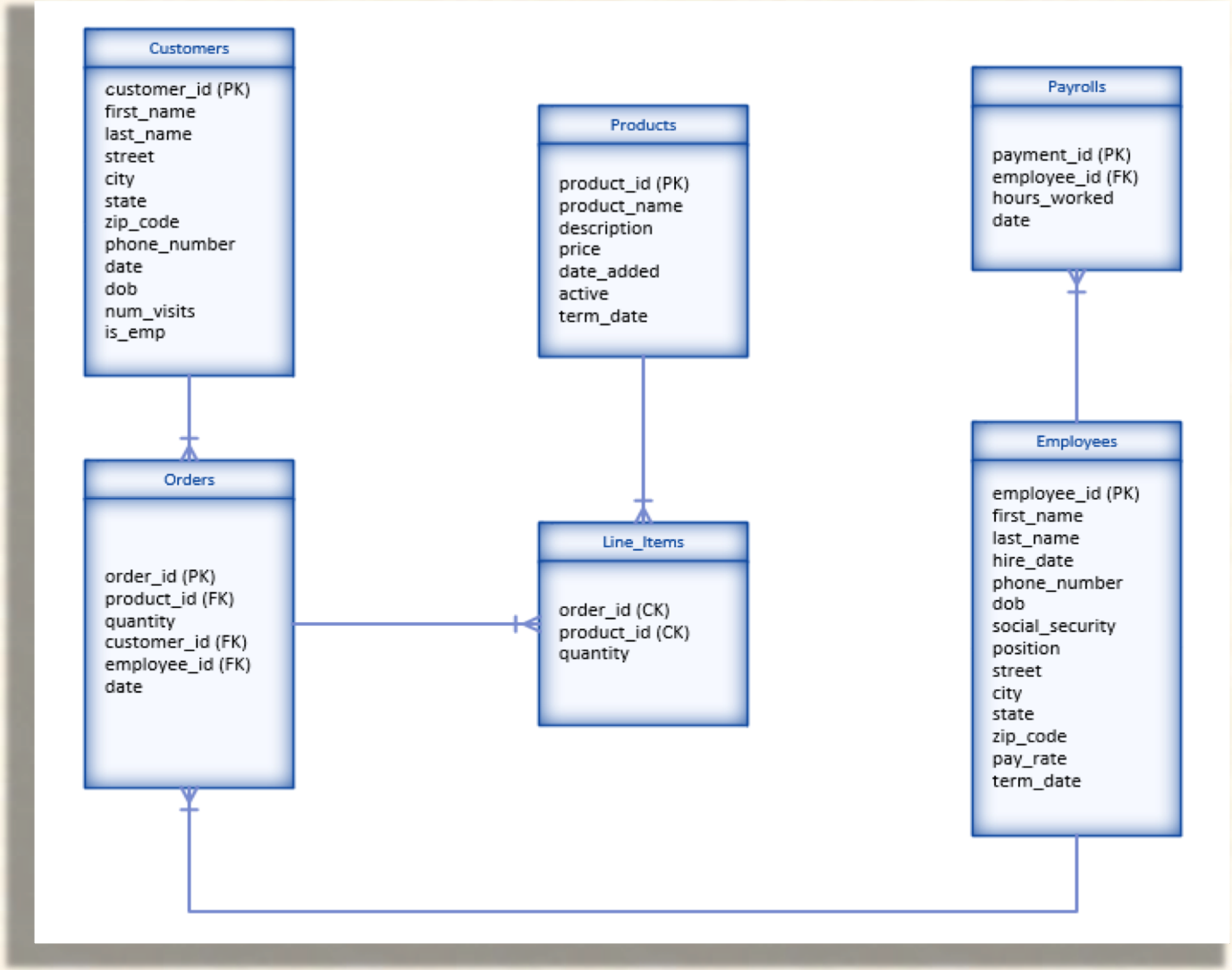
Alejandro's is an Italian restaurant specializing on the finest Italian dishes and desserts, we have a great location and a great environment to make your visits a pleasant experience. Bring your family and enjoy the delicious food served at our restaurants.

parallelo

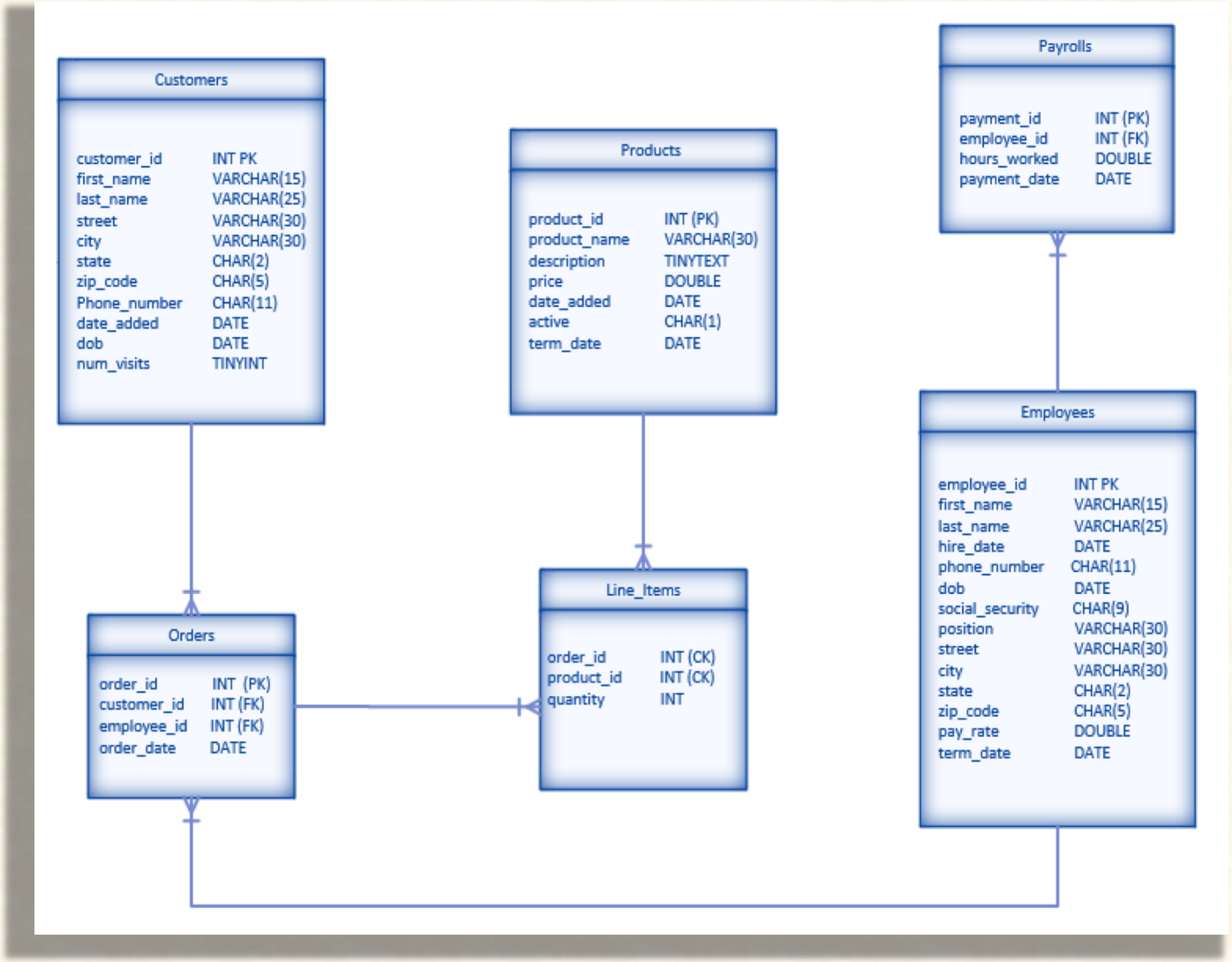
Mission Statement

Our mission is to provide quality foods to our clients and a home like environment for the families that visit our locations. We thrive to be the first choice when good healthy and delicious food is desired by our customers.

ERD


Customers
customer_id (PK)
first_name
last_name
street
city
state
zip_code
phone_number
date
dob
num_visits
is_emp
Products
product_id (PK)
product_name
description
price
date_added
active
term_date
Payrolls
payment_id (PK)
employee_id (FK)
hours_worked
date
Orders
order_id (PK)
product_id (FK)
quantity
customer_id (FK)
employee_id (FK)
date
Line_Items
order_id (CK)
product_id (CK)
quantity
Employees
employee_id (PK)
first_name
last_name
hire_date
phone_number
dob
social_security
position
street
city
state
zip_code
pay_rate
term_date

Normalization



Company Business Rules

Must work at least 30 hours per week and no more than 50:

```
SELECT e.first_name, e.last_name, p.hours_worked FROM
Employees e INNER JOIN Payrolls p
ON e.employee_id = p.employee_id
WHERE p.hours_worked >= 30 AND p.hours_worked <= 50
AND p.payment_date  BETWEEN '2018-02-01' AND '2018-02-28';
```

SQL Statement: SELECT e.first_name, e.last_n
AND p.hours_worked <= 50 AND p.payment_da

Output:

first_name	last_name	hours_worked
Rick	James	44
Tony	Smith	43
Walter	White	49
Denise	Jones	43
Sally	Webber	48
Ron	Swanson	35
Bob	Vance	40

The employee with most services gets a 15% bonus pay:

```
SELECT COUNT(order_id), first_name, last_name
FROM Employees INNER JOIN Orders
ON Employees.employee_id = Orders.employee_id
GROUP BY first_name
HAVING COUNT(Orders.order_id) >= 1;
```

SQL Statement: SELECT COUNT(order_id), first_
HAVING COUNT(Orders.order_id) >= 1;

Output:

COUNT(order_id)	first_name	last_name
9	Ron	Swanson
13	Sally	Webber
13	Tim	Montgomery
4	Tony	Black

Database Tables Creation

```
CREATE TABLE Customers
(
customer_id     INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
first_name      VARCHAR(15),
last_name       VARCHAR(25),
street          VARCHAR(30),
city            VARCHAR(30),
state           CHAR(2),
zip_code        CHAR(5),
phone_number    CHAR(11),
date_added      DATE,
dob             DATE,
num_visits      TINYINT
)
AUTO_INCREMENT = 1000 ENGINE = INNODB;
```

```
CREATE TABLE Employees
(
employee_id     INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
first_name      VARCHAR(15),
last_name       VARCHAR(25),
hire_date       DATE,
phone_number    CHAR(11),
dob             DATE,
social_security CHAR(9),
position        VARCHAR(30),
street          VARCHAR(30),
city            VARCHAR(30),
state           CHAR(2),
zip_code        CHAR(5),
pay_rate        DOUBLE,
term_date       DATE
)
AUTO_INCREMENT = 1000 ENGINE = INNODB;
```

Table Data Insert

```
INSERT INTO Employees VALUES

(NULL, 'Rick',  'James',  '2015-12-17', 19085310047, '1987-05-29', 555889999, 'Cook',  '45 Oak St', 'Pittston', 'PA', 18640, 12.95, NULL),

(NULL, 'Tony', 'Smith', '2014-04-20', 15709991478, '1997-05-29', 666229459, 'Dishwasher', '305 William St', 'Pittston', 'PA', 18640, 8.50,  NULL),

(NULL, 'Walter', 'White', '2017-07-02', 15706234897, '1967-11-13', 123456789, 'Manager', '444 Welles St', 'Forty Fort', 'PA', 18704, 18.99, NULL),

(NULL, 'Peter', 'Pan', '2016-01-05', 15705552829, '1992-08-22', 987654321, 'Cook', '111 Arch St', 'Edwardsville', 'PA', 18704, 12.95, NULL);
```

SQL Statement: SELECT * FROM Employees;

Output:

employee_id	first_name	last_name	hire_date	phone_number	dob	social_security	position	street	city	state	zip_code	pay_rate	term_date
1000	Rick	James	2015-12-17	19085310047	1987-05-29	555889999	Cook	45 Oak St	Pittston	PA	18640	12.95	null
1001	Tony	Smith	2014-04-20	15709991478	1997-05-29	666229459	Dishwasher	305 William St	Pittston	PA	18640	8.5	null
1002	Walter	White	2017-07-02	15706234897	1967-11-13	123456789	Manager	444 Welles St	Forty Fort	PA	18704	18.99	null
1003	Peter	Pan	2016-01-05	15705552829	1992-08-22	987654321	Cook	111 Arch St	Edwardsville	PA	18704	12.95	null
1004	Juli	Myers	2016-10-15	15708784444	1990-07-02	519763482	Hostess	737 Carey Ave	Wilkes-Barre	PA	18703	10	null
1005	Denise	Jones	2010-06-12	15701226549	1985-01-24	963852741	Waitress	25 Hazel St	Wilkes-Barre	PA	18703	12.85	null
1006	Sally	Webber	2010-09-09	15708884561	1988-03-11	147369258	Waitress	32 Water St	Dallas	PA	18612	12.85	null
1007	Ron	Swanson	2011-04-22	15702975581	1994-09-08	951357456	Waiter	159 Carle St	Kingston	PA	18704	12.85	null
1008	Tim	Montgomery	2012-02-18	15709964301	1989-03-23	966488235	Waiter	786 Rutter Ave	Kingston	PA	18704	12.85	null
1009	Bob	Vance	2015-01-11	15700008899	1991-04-17	115663247	Cook	150 Owen St	Swoyersville	PA	18704	12.95	null
1010	Sarah	Thomas	2012-12-12	15709996543	1991-06-27	758946223	Cook	167 Oak St	Pittston	PA	18640	12.95	2014-04-13
1011	Barry	White	2016-02-20	15708216644	1980-06-19	884992517	Dishwasher	36 Mack St	Plains	PA	18702	8.5	2018-02-17
1012	Tony	Black	2015-09-02	15707713569	1993-10-13	551174963	Waiter	648 Henry St	Plains	PA	18704	2.85	2016-10-07
1013	John	Jones	2017-12-16	15704568879	1994-11-12	2633458	Dishwasher	26 Parry St	Luzerne	PA	18709	8.5	null
1014	Debora	Johnson	2010-04-16	15704545229	1985-06-16	548789631	Waitress	451 Green St	Drums	PA	18709	12.95	null

Table Data Insert

```
INSERT INTO Customers VALUES

(NULL, 'George', 'Smith', '158 Maple St', 'Plains Township', 'PA', '18702', '5708566321', '2010-06-22', '1981-09-20', 3),

(NULL, 'Lucas', 'Brown', '211 Robert St', 'Nanticoke', 'PA', '18634', '5701245785', '2003-08-01', '1975-10-14', 3),

(NULL, 'Jim', 'Krupilis', '125 W Church St', 'Nanticoke', 'PA', '18634', '5707896352', '2009-09-12', '1994-03-26', 4),

(NULL, 'Laura', 'Andrasko', '233 Delaware St', 'Scranton', 'PA', '18505', '5707413254', '2018-02-24', '1979-05-25', 8);
```

SQL Statement: SELECT * FROM Customers;

Output:

customer_id	first_name	last_name	street	city	state	zip_code	phone_number	date_added	dob	num_visits	is_emp
1000	George	Smith	158 Maple St	Plains Township	PA	18702	5708566321	2010-06-22	1981-09-20	3	null
1001	Lucas	Brown	211 Robert St	Nanticoke	PA	18634	5701245785	2003-08-01	1975-10-14	3	null
1002	Jim	Krupilis	125 W Church St	Nanticoke	PA	18634	5707896352	2009-09-12	1994-03-26	4	null
1003	Laura	Andrasko	233 Delaware St	Scranton	PA	18505	5707413254	2018-02-24	1979-05-25	8	null
1004	Justin	Macko	45 E Thomas St	Plains Township	PA	18702	5704521234	2014-04-16	1970-10-20	12	null
1005	Joseph	Smith	260 Muir Ave	Hazleton	PA	18201	5701237589	1999-02-10	1984-09-09	8	null
1006	Anderson	McCutchen	12 Pierce St	Kingston	PA	18704	5706844322	2012-07-21	1994-02-05	3	null
1007	Mathew	Florio	45 Matson Ave	Plains Township	PA	18702	5701477856	2002-06-19	1960-01-01	5	null
1008	Aaron	Graham	56 Kidder St	Wilkes Barre	PA	18711	5703692145	2006-03-26	1965-06-11	10	null
1009	Michelle	Kinney	354 Spring St	Dallas	PA	18612	5704784563	1999-04-30	1977-07-19	11	null
1010	Dan	Johnson	45 Laflin Rd	Plains Township	PA	18705	5558566321	2013-01-22	1990-08-15	15	null
1011	Maria	Rogers	111 Main St	Sugar Notch	PA	18706	5554786321	2016-08-11	1996-04-22	6	null
1012	Othello	Palermo	33 S Kennedy Dr	McAdoo	PA	18237	5708566399	2013-11-30	1975-11-23	14	null
1013	John	Kowalsky	456 Hunter St	Tamaqua	PA	18252	6988545321	2009-10-13	1988-07-01	9	null
1014	Nancy	Adorno	165 Centre St	Sheppton	PA	18248	5703846321	2018-02-22	1960-12-20	6	null
1015	Rick	James	45 Oak St	Pittston	PA	18640	19085310047	2015-12-29	1987-05-29	5	Y
1016	Tony	Smith	305 William St	Pittston	PA	18640	15709991478	2014-05-15	1997-05-29	27	Y
1017	Walter	White	444 Welles St	Forty Fort	PA	18704	15706234897	2017-07-02	1967-11-13	89	Y

Simple Queries

Customers with less than 5 visits to the restaurant:

```
SELECT first_name, last_name FROM Customers
WHERE num_visits < 5;
```

SQL Statement: SELECT first_name, last_na

Output:

first_name	last_name
George	Smith
Lucas	Brown
Jim	Krupilis
Anderson	McCutchen

Average pay rate for employees:

```
SELECT AVG(pay_rate) AS AveragePay
FROM Employees;
```

SQL Statement: SELECT AVG(pay_rate)

Output:

AveragePay
11.565999999999997

Complex Queries

Show products that cost over $5.00 and were added to the menu before 2007:

```
SELECT DISTINCT product_name, price,
date_added
FROM Products INNER JOIN Line_Items
ON Products.product_id =
Line_Items.product_id
WHERE price > 5.00 AND date_added <
‘2007-01-01’
ORDER BY date_added;
```

SQL Statement: SELECT DISTINCT product
Line_Items.product_id WHERE price > 5.00 A

Output:

product_name	price	date_added
Veggie Pizza	7.95	1999-06-02
Pepperoni Pizza	8.25	1999-08-02
Quattro Formaggio Pizza	7.45	1999-12-06
Flag Pizza	7.95	1999-12-06
Caesar Salad	12.99	2006-11-09

Show the product id, name, quantities sold and price of the products from our menu in the month of April:

```
SELECT
Products.product_id AS 'Product ID',
Products.product_name AS 'Product Name',
SUM(Line_Items.quantity) AS 'Quantities Sold',
Products.price AS 'Price'
FROM Line_Items INNER JOIN Products ON Products.product_id = Line_Items.product_id
INNER JOIN Orders ON Orders.order_id = Line_Items.order_id
WHERE Line_Items.product_id = Line_Items.product_id AND Orders.order_date > '2018-03-31' AND
Orders.order_date < '2018-04-30‘ GROUP BY Products.product_name;
```

SQL Statement: SELECT Products.product_id AS 'Product ID'
Products.price AS 'Price' FROM Line_Items INNER JOIN Prod
Orders.order_id = Line_Items.order_id WHERE Line_Items.pro
Orders.order_date < '2018-04-30' GROUP BY Products.product_

Output:

Product ID	Product Name	Quantities Sold	Price
1016	Bottled Water	3	1.75
1001	Caesar Salad	2	12.99
1002	Chicken Parm	6	15.99
1007	Flag Pizza	5	7.95
1020	Gelato	6	1.75
1003	Italian Sausage	10	12.99
1009	Pepperoni Pizza	2	8.25

Customer Invoice

Customer invoice:

```
SELECT o.order_id, l.product_id, l.quantity, p.price, (p.price * l.quantity) AS "Line Item Total",
(SELECT SUM(p.price * l.quantity) FROM Orders o, Line_Items l, Products p
WHERE o.order_id = l.order_id AND l.product_id = p.product_id AND o.order_id = 10008) AS "Grand Total"
FROM Orders o, Line_Items l, Products p
WHERE o.order_id = l.order_id AND l.product_id = p.product_id AND o.order_id = 10008;
```

SQL Statement: SELECT o.order_id, l.product_id, l.quantity, p.price, (
Orders o, Line_Items l, Products p WHERE o.order_id = l.order_id ANI
Orders o, Line_Items l, Products p WHERE o.order_id = l.order_id ANI

Output:

order_id	product_id	quantity	price	Line Item Total	Grand Total
10008	1003	2	12.99	25.98	52.68
10008	1006	1	7.45	7.45	52.68
10008	1016	1	1.75	1.75	52.68
10008	1019	5	3.5	17.5	52.68

Queries with Sub-Query

Percentage of employees that are listed as customers:

```
SELECT
(COUNT(*)-(SELECT COUNT(*) FROM Customers WHERE is_emp = 'Y'))/COUNT(*)
AS '% Of Employees as Customers' FROM Customers;
```

SQL Statement: SELECT (COUNT(FROM Customers

Output:

% Of Employees as Customers
0.8333

Show products where the price is higher than the average price of our products:

```
SELECT product_name, price
FROM Products WHERE price >= (SELECT AVG(price)
FROM Products);
```

SQL Statement: SELECT product_nar

Output:

product_name	price
Caprese Salad	10.99
Caesar Salad	12.99
Chicken Parm	15.99
Italian Sausage	12.99
Spaghetti Pomodoro	14.99

Calculation Queries

Gross pay amount by employee with overtime included:

```
SELECT Employees.employee_id, Employees.first_name, Employees.last_name,
(((Payrolls.hours_worked-40) * (Employees.pay_rate/2)) + (40 * Employees.pay_rate)) AS 'Gross pay'
FROM Payrolls INNER JOIN Employees ON Payrolls.employee_id = Employees.employee_id
WHERE
Payrolls.hours_worked > 40;
```

SQL Statement: SELECT Employees.employee_id
(40 * Employees.pay_rate)) AS 'Gross pay' FROM F
Payrolls.hours_worked > 40;

Output:

employee_id	first_name	last_name	Gross pay
1003	Peter	Pan	550.375
1006	Sally	Webber	526.85
1007	Ron	Swanson	571.825
1009	Bob	Vance	524.475
1000	Rick	James	530.95
1005	Denise	Jones	610.375
1007	Ron	Swanson	526.85
1000	Rick	James	543.9
1001	Tony	Smith	352.75
1002	Walter	White	845.055
1005	Denise	Jones	533.275
1006	Sally	Webber	565.4

Calculation Queries

How many and how much of a product has been sold:

```
SELECT
p.product_id, p.product_name,
(SELECT SUM(quantity) FROM Line_Items WHERE product_id = 1003) AS 'Quantity Sold',
(SELECT SUM(quantity) FROM Line_Items WHERE product_id = 1003) * p.price AS 'Total'
FROM
Line_Items l INNER JOIN Products p ON  l.product_id = p.product_id AND l.product_id = 1003
INNER JOIN Orders o ON o.order_id = l.order_id AND order_date < '2018-04-30'
WHERE l.product_id = 1003 GROUP BY product_name;
```

SQL Statement: SELECT p.product_id, p.product_n
(SELECT SUM(quantity) FROM Line_Items WHER
= p.product_id AND l.product_id = 1003 INNER JOI
GROUP BY product_name;

Output:

product_id	product_name	Quantity Sold	Total
1003	Italian Sausage	10	129.9

CIS – 148 – 001

Spring 2018

Instructor: Earl Weidner

DATABASE PROJECT – GROUP 3

Distinguished Members:

Hector Cedeno

Jim Loomis

Jose Ramirez

Edwin Aguilar

Thank You!