ORACLE Academy academy.oracle.com

Database Design Project

Oracle Baseball League Store Database

Project Scenario:

You are a small consulting company specializing in database development. You have just been awarded the contract to develop a data model for a database application system for a small retail store called Oracle Baseball League (OBL).

The Oracle Baseball League store serves the entire surrounding community selling baseball kit. The OBL has two types of customer, there are individuals who purchase items like balls, cleats, gloves, shirts, screen printed t-shirts, and shorts. Additionally customers can represent a team when they purchase uniforms and equipment on behalf of the team.

Teams and individual customers are free to purchase any item from the inventory list, but teams get a discount on the list price depending on the number of players. When a customer places an order we record the order items for that order in our database.

OBL has a team of three sales representatives that officially only call on teams but have been known to handle individual customer complaints.

Section 6 Lesson 9 Exercise 1: Joining Tables Using JOIN

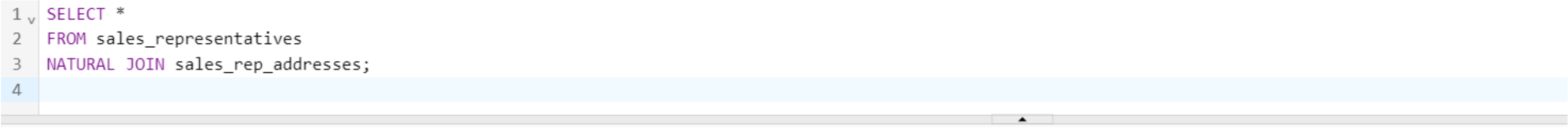
Write SELECT Statements Using Data From Multiple Tables Using Equijoins and Non-Equijoins (S6L9 Objective 1)

In this exercise you will write SELECT statements to access data from more than one table.

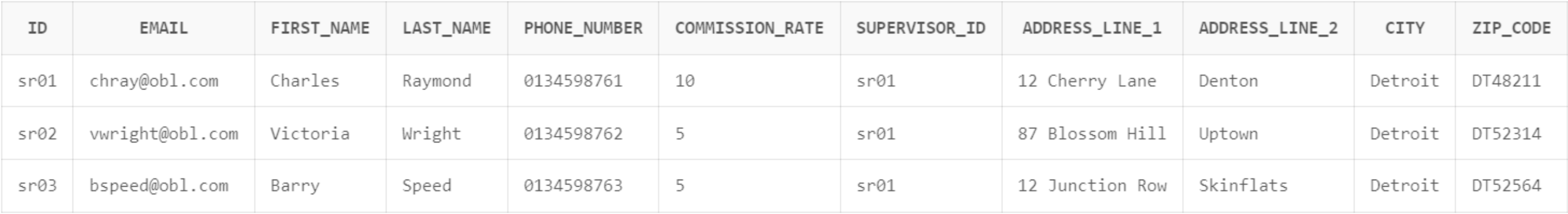
NAME: PANG ZHI XUAN
MATRIC NO: A22SC0466

Part 1: Creating Natural Joins.

1. Display all of the information about sales representatives and their addresses using a natural join.

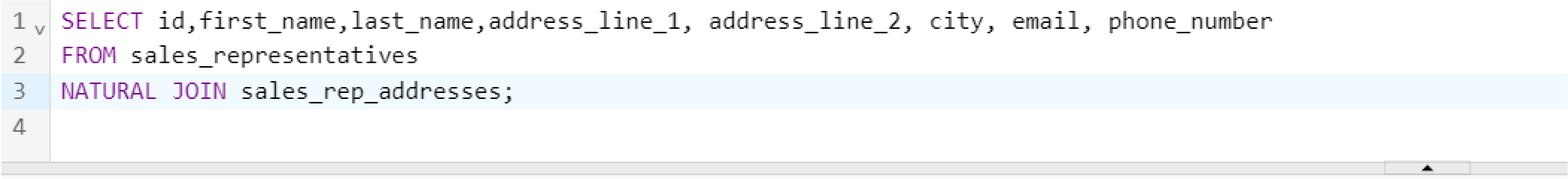


```
SELECT *
FROM sales_representatives
NATURAL JOIN sales_rep_addresses;
```

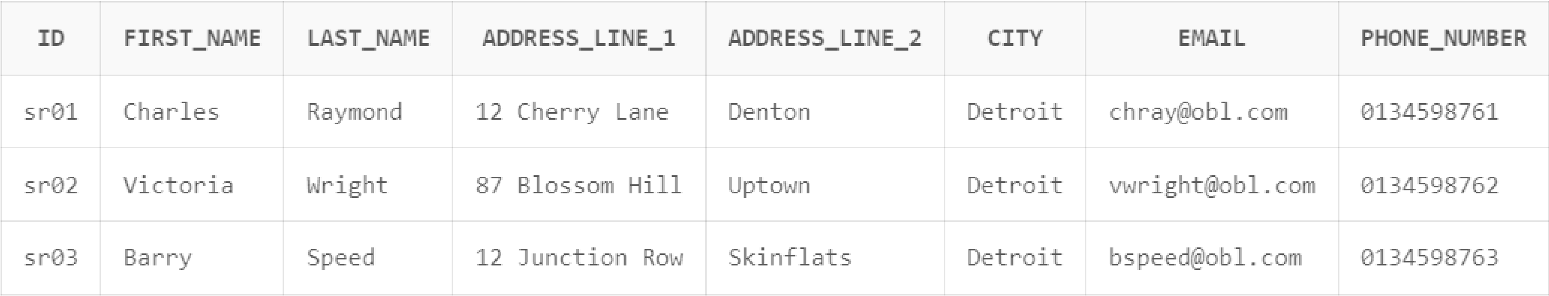


ID	EMAIL	FIRST_NAME	LAST_NAME	PHONE_NUMBER	COMMISSION_RATE	SUPERVISOR_ID	ADDRESS_LINE_1	ADDRESS_LINE_2	CITY	ZIP_CODE
sr01	chray@obl.com	Charles	Raymond	0134598761	10	sr01	12 Cherry Lane	Denton	Detroit	DT48211
sr02	vwright@obl.com	Victoria	Wright	0134598762	5	sr01	87 Blossom Hill	Uptown	Detroit	DT52314
sr03	bspeed@obl.com	Barry	Speed	0134598763	5	sr01	12 Junction Row	Skinflats	Detroit	DT52564

2. Adapt the query from the previous question to only show the id, first name, last name, address line 1, address line 2, city, email and phone_number for the sales representatives.



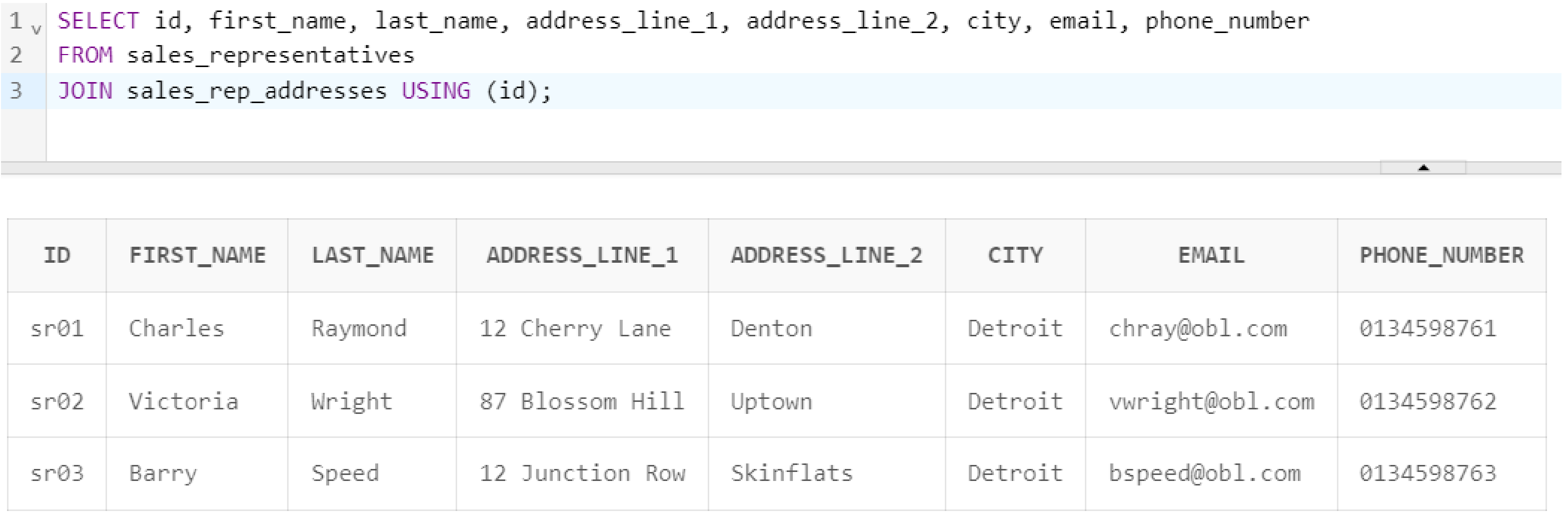
```
SELECT id,first_name,last_name,address_line_1, address_line_2, city, email, phone_number
FROM sales_representatives
NATURAL JOIN sales_rep_addresses;
```



ID	FIRST_NAME	LAST_NAME	ADDRESS_LINE_1	ADDRESS_LINE_2	CITY	EMAIL	PHONE_NUMBER
sr01	Charles	Raymond	12 Cherry Lane	Denton	Detroit	chray@obl.com	0134598761
sr02	Victoria	Wright	87 Blossom Hill	Uptown	Detroit	vwright@obl.com	0134598762
sr03	Barry	Speed	12 Junction Row	Skinflats	Detroit	bspeed@obl.com	0134598763

Part 2: Creating Joins with the USING Clause

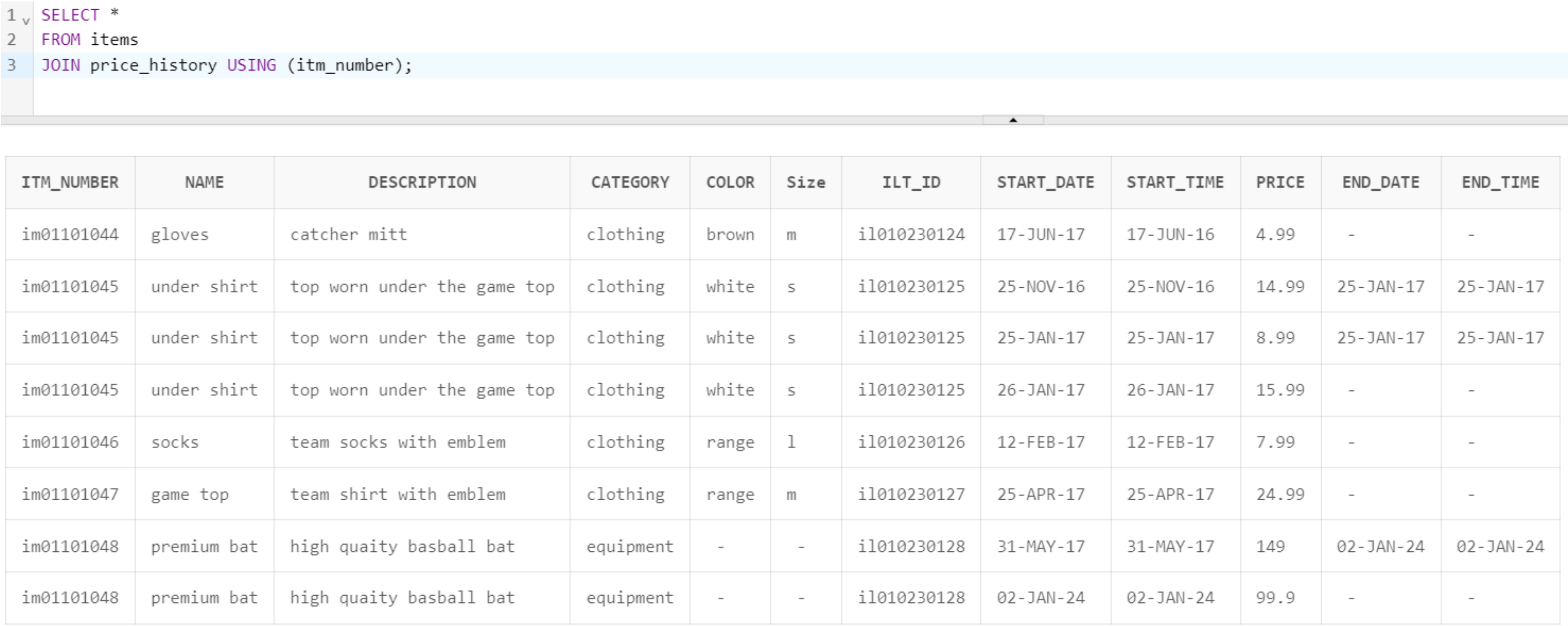
1. Adapt the previous query answer to use the USING clause instead of a natural join.



```sql
SELECT id, first_name, last_name, address_line_1, address_line_2, city, email, phone_number
FROM sales_representatives
JOIN sales_rep_addresses USING (id);
```

ID	FIRST_NAME	LAST_NAME	ADDRESS_LINE_1	ADDRESS_LINE_2	CITY	EMAIL	PHONE_NUMBER
sr01	Charles	Raymond	12 Cherry Lane	Denton	Detroit	chray@obl.com	0134598761
sr02	Victoria	Wright	87 Blossom Hill	Uptown	Detroit	vwright@obl.com	0134598762
sr03	Barry	Speed	12 Junction Row	Skinflats	Detroit	bspeed@obl.com	0134598763

2. Display all of the information about items and their price history by joining the items and price_history tables.

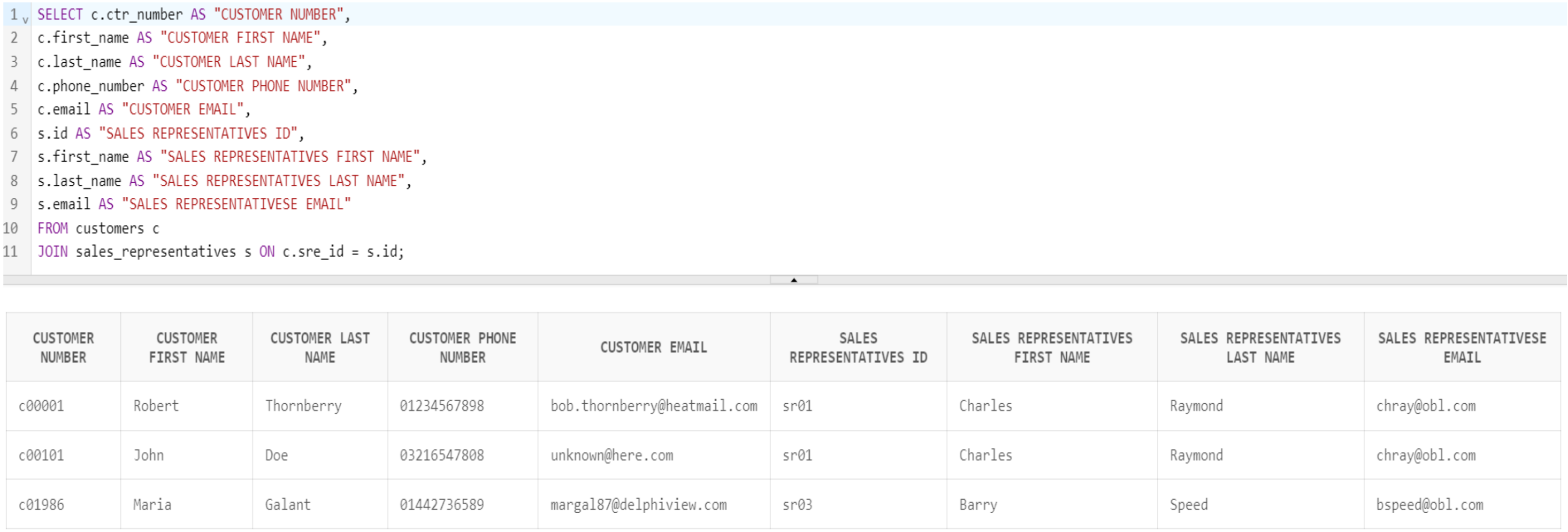


```sql
SELECT *
FROM items
JOIN price_history USING (itm_number);
```

ITM_NUMBER	NAME	DESCRIPTION	CATEGORY	COLOR	Size	ILT_ID	START_DATE	START_TIME	PRICE	END_DATE	END_TIME
im01101044	gloves	catcher mitt	clothing	brown	m	il010230124	17-JUN-17	17-JUN-16	4.99	-	-
im01101045	under shirt	top worn under the game top	clothing	white	s	il010230125	25-NOV-16	25-NOV-16	14.99	25-JAN-17	25-JAN-17
im01101045	under shirt	top worn under the game top	clothing	white	s	il010230125	25-JAN-17	25-JAN-17	8.99	25-JAN-17	25-JAN-17
im01101045	under shirt	top worn under the game top	clothing	white	s	il010230125	26-JAN-17	26-JAN-17	15.99	-	-
im01101046	socks	team socks with emblem	clothing	range	l	il010230126	12-FEB-17	12-FEB-17	7.99	-	-
im01101047	game top	team shirt with emblem	clothing	range	m	il010230127	25-APR-17	25-APR-17	24.99	-	-
im01101048	premium bat	high quaity basball bat	equipment	-	-	il010230128	31-MAY-17	31-MAY-17	149	02-JAN-24	02-JAN-24
im01101048	premium bat	high quaity basball bat	equipment	-	-	il010230128	02-JAN-24	02-JAN-24	99.9	-	-

Part 3: Creating Joins with the ON Clause

1. Use an ON clause to join the customer and sales representative table so that you display the customer number, customer fist name, customer last name, customer phone number, customer email, sales representative id, sales representative first name, sales representative last name and sales representative email. You will need to use a table alias in your answer as both tables have columns with the same name.

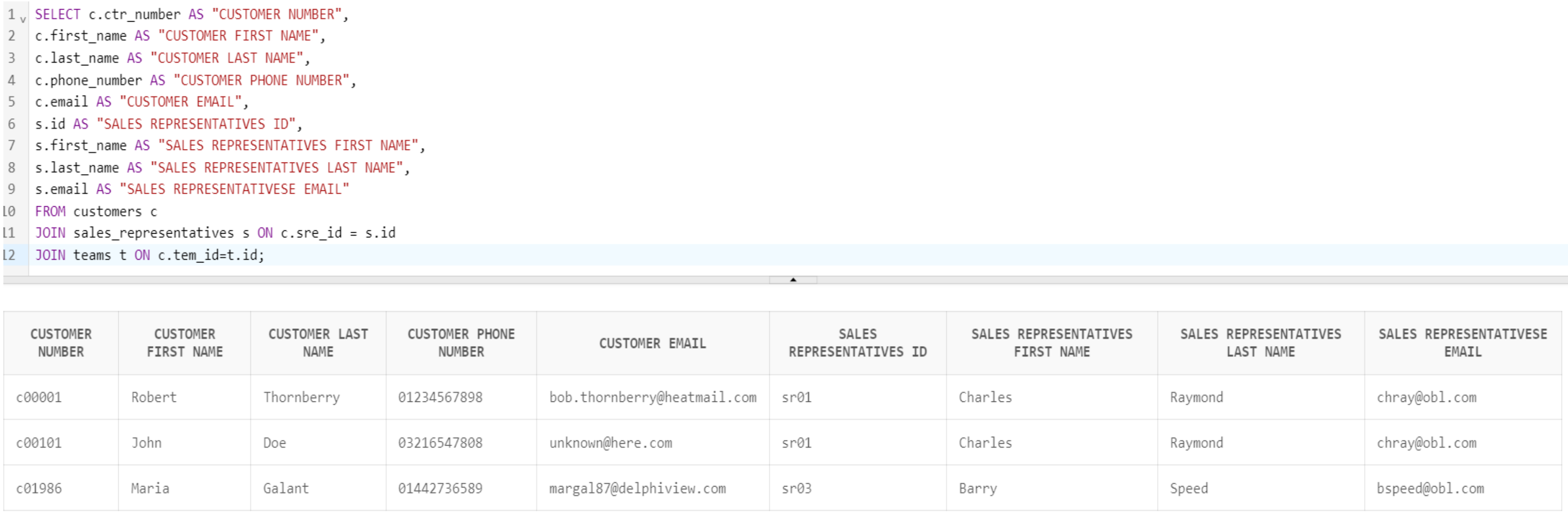


```sql
SELECT c.ctr_number AS "CUSTOMER NUMBER",
c.first_name AS "CUSTOMER FIRST NAME",
c.last_name AS "CUSTOMER LAST NAME",
c.phone_number AS "CUSTOMER PHONE NUMBER",
c.email AS "CUSTOMER EMAIL",
s.id AS "SALES REPRESENTATIVES ID",
s.first_name AS "SALES REPRESENTATIVES FIRST NAME",
s.last_name AS "SALES REPRESENTATIVES LAST NAME",
s.email AS "SALES REPRESENTATIVESE EMAIL"
FROM customers c
JOIN sales_representatives s ON c.sre_id = s.id;
```

CUSTOMER NUMBER	CUSTOMER FIRST NAME	CUSTOMER LAST NAME	CUSTOMER PHONE NUMBER	CUSTOMER EMAIL	SALES REPRESENTATIVES ID	SALES REPRESENTATIVES FIRST NAME	SALES REPRESENTATIVES LAST NAME	SALES REPRESENTATIVESE EMAIL
c00001	Robert	Thornberry	01234567898	bob.thornberry@heatmail.com	sr01	Charles	Raymond	chray@obl.com
c00101	John	Doe	03216547808	unknown@here.com	sr01	Charles	Raymond	chray@obl.com
c01986	Maria	Galant	01442736589	margal87@delphiview.com	sr03	Barry	Speed	bspeed@obl.com

Part 4- Creating Three-Way Joins with the ON Clause

1. Using the answer to Task 3 add a join that will allow the team name that the customer represents to be included in the results.

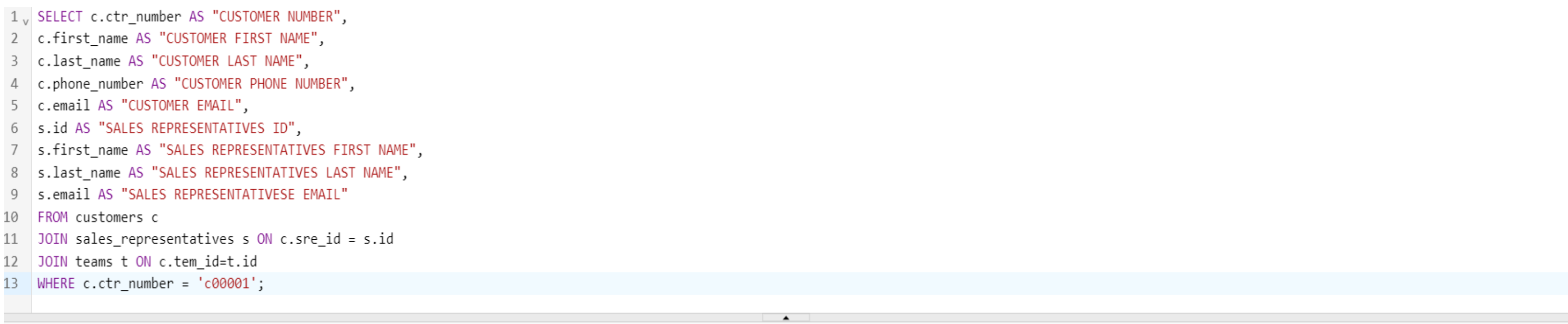


```sql
SELECT c.ctr_number AS "CUSTOMER NUMBER",
c.first_name AS "CUSTOMER FIRST NAME",
c.last_name AS "CUSTOMER LAST NAME",
c.phone_number AS "CUSTOMER PHONE NUMBER",
c.email AS "CUSTOMER EMAIL",
s.id AS "SALES REPRESENTATIVES ID",
s.first_name AS "SALES REPRESENTATIVES FIRST NAME",
s.last_name AS "SALES REPRESENTATIVES LAST NAME",
s.email AS "SALES REPRESENTATIVESE EMAIL"
FROM customers c
JOIN sales_representatives s ON c.sre_id = s.id
JOIN teams t ON c.tem_id=t.id;
```

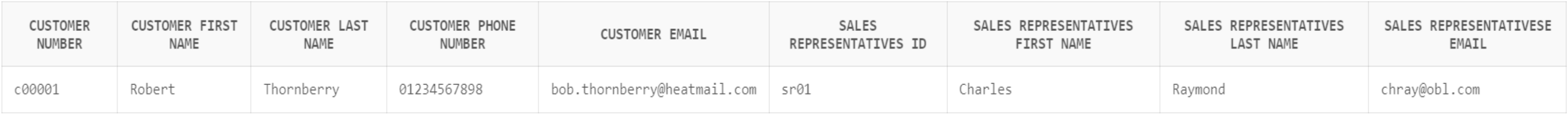
CUSTOMER NUMBER	CUSTOMER FIRST NAME	CUSTOMER LAST NAME	CUSTOMER PHONE NUMBER	CUSTOMER EMAIL	SALES REPRESENTATIVES ID	SALES REPRESENTATIVES FIRST NAME	SALES REPRESENTATIVES LAST NAME	SALES REPRESENTATIVESE EMAIL
c00001	Robert	Thornberry	01234567898	bob.thornberry@heatmail.com	sr01	Charles	Raymond	chray@obl.com
c00101	John	Doe	03216547808	unknown@here.com	sr01	Charles	Raymond	chray@obl.com
c01986	Maria	Galant	01442736589	margal87@delphiview.com	sr03	Barry	Speed	bspeed@obl.com

Part 5: Applying Additional Conditions to a Join

1. Using the answer to Task 4 add an additional condition to only show the results for the customer that has the number - c00001.



```sql
SELECT c.ctr_number AS "CUSTOMER NUMBER",
c.first_name AS "CUSTOMER FIRST NAME",
c.last_name AS "CUSTOMER LAST NAME",
c.phone_number AS "CUSTOMER PHONE NUMBER",
c.email AS "CUSTOMER EMAIL",
s.id AS "SALES REPRESENTATIVES ID",
s.first_name AS "SALES REPRESENTATIVES FIRST NAME",
s.last_name AS "SALES REPRESENTATIVES LAST NAME",
s.email AS "SALES REPRESENTATIVESE EMAIL"
FROM customers c
JOIN sales_representatives s ON c.sre_id = s.id
JOIN teams t ON c.tem_id=t.id
WHERE c.ctr_number = 'c00001';
```

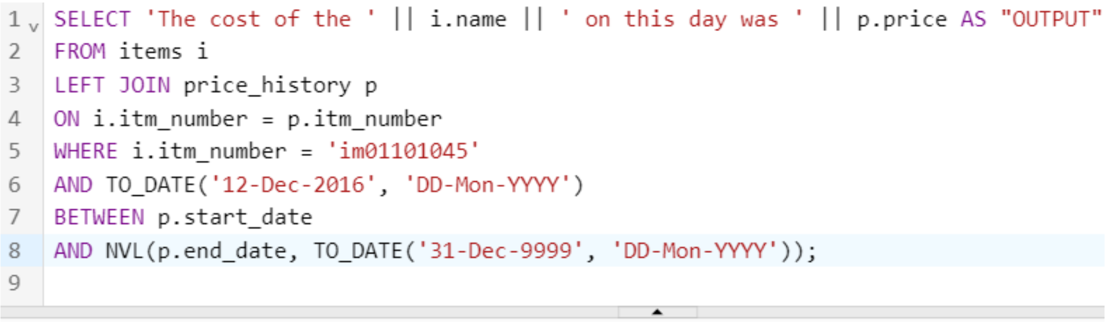


CUSTOMER NUMBER	CUSTOMER FIRST NAME	CUSTOMER LAST NAME	CUSTOMER PHONE NUMBER	CUSTOMER EMAIL	SALES REPRESENTATIVES ID	SALES REPRESENTATIVES FIRST NAME	SALES REPRESENTATIVES LAST NAME	SALES REPRESENTATIVESE EMAIL
c00001	Robert	Thornberry	01234567898	bob.thornberry@heatmail.com	sr01	Charles	Raymond	chray@obl.com

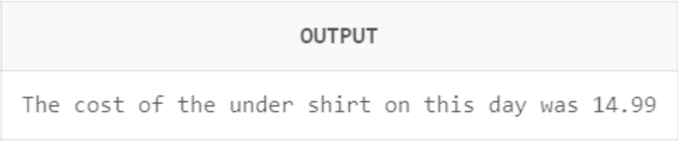
Part 6: Retrieving Records with Nonequijoins

1. Write a query that will display name and cost of the item with the number im01101045 on the 12th of December 2016. The output of the query should look like this:

The cost of the under shirt on this day was 14.99



```sql
SELECT 'The cost of the ' || i.name || ' on this day was ' || p.price AS "OUTPUT"
FROM items i
LEFT JOIN price_history p
ON i.itm_number = p.itm_number
WHERE i.itm_number = 'im01101045'
AND TO_DATE('12-Dec-2016', 'DD-Mon-YYYY')
BETWEEN p.start_date
AND NVL(p.end_date, TO_DATE('31-Dec-9999', 'DD-Mon-YYYY'));
```



OUTPUT
The cost of the under shirt on this day was 14.99