

Lab 4

Name: Tam Jia Hao

Matric No: A22EC0106

Course Name: Database

Course Code: SECD2523

Section: 06

Lecturer: Dr Izyan Izzati binti Kamsani

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.

Copyright © 2020, Oracle and/or its affiliates. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

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;
```

The screenshot shows the APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. A search bar and user profile 'TJ TAM JIA HAO' are on the right. The 'SQL Commands' panel shows the query: `SELECT * FROM sales_representatives NATURAL JOIN sales_rep_addresses;`. The 'Results' panel displays a table with 11 columns: ID, EMAIL, FIRST_NAME, LAST_NAME, PHONE_NUMBER, COMMISSION_RATE, SUPERVISOR_ID, ADDRESS_LINE_1, ADDRESS_LINE_2, CITY, and ZIP_CODE. Three rows of data are shown, corresponding to sales representatives sr01, sr02, and sr03. The status bar at the bottom indicates '3 rows returned in 0.05 seconds' and provides a 'Download' link.

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

- 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;
```

The screenshot shows the APEX SQL Workshop interface. The SQL command is: `SELECT id, first_name, last_name, address_line_1, address_line_2, city, email, phone_number FROM sales_representatives NATURAL JOIN sales_rep_addresses;` The results are displayed in a table with 8 columns: ID, FIRST_NAME, LAST_NAME, ADDRESS_LINE_1, ADDRESS_LINE_2, CITY, EMAIL, and PHONE_NUMBER. The results show 3 rows of data for sales representatives Charles Raymond, Victoria Wright, and Barry Speed.

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

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

```
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);
```

The screenshot shows the APEX SQL Workshop interface. The SQL command is: `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);` The results are displayed in a table with 8 columns: ID, FIRST_NAME, LAST_NAME, ADDRESS_LINE_1, ADDRESS_LINE_2, CITY, EMAIL, and PHONE_NUMBER. The results show 3 rows of data for sales representatives Charles Raymond, Victoria Wright, and Barry Speed.

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.

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

APEX App Builder SQL Workshop Team Development Gallery

Search

SQL Commands Schema WKSP_DATABASEJIAHAOT/

Language SQL Rows 10 Clear Command Find Tables Save Run

```
1 SELECT *  
2 FROM items JOIN price_history  
3 USING (itm_number);
```

Results Explain Describe Saved SQL History

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	06/17/2017	06/17/2016	4.99	-	-
im01101045	under shirt	top worn under the game top	clothing	white	s	il010230125	11/25/2016	11/25/2016	14.99	01/25/2017	01/25/2017
im01101045	under shirt	top worn under the game top	clothing	white	s	il010230125	01/25/2017	01/25/2017	8.99	01/25/2017	01/25/2017
im01101045	under shirt	top worn under the game top	clothing	white	s	il010230125	01/26/2017	01/26/2017	15.99	-	-
im01101046	socks	team socks with emblem	clothing	range	l	il010230126	02/12/2017	02/12/2017	7.99	-	-
im01101047	game top	team shirt with emblem	clothing	range	m	il010230127	04/25/2017	04/25/2017	24.99	-	-
im01101048	premium bat	high quaity basball bat	equipment	-	-	il010230128	05/31/2017	05/31/2017	149	11/09/2023	11/09/2023

jiahao0417@gmail.com databasejiahaotam en Copyright © 1999, 2023, Oracle and/or its affiliates. Oracle APEX 23.21

APEX App Builder SQL Workshop Team Development Gallery

Search

SQL Commands Schema WKSP_DATABASEJIAHAOT/

Language SQL Rows 10 Clear Command Find Tables Save Run

```
1 SELECT *  
2 FROM items JOIN price_history  
3 USING (itm_number);
```

Results Explain Describe Saved SQL History

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	06/17/2017	06/17/2016	4.99	-	-
im01101045	under shirt	top worn under the game top	clothing	white	s	il010230125	11/25/2016	11/25/2016	14.99	01/25/2017	01/25/2017
im01101045	under shirt	top worn under the game top	clothing	white	s	il010230125	01/25/2017	01/25/2017	8.99	01/25/2017	01/25/2017
im01101045	under shirt	top worn under the game top	clothing	white	s	il010230125	01/26/2017	01/26/2017	15.99	-	-
im01101046	socks	team socks with emblem	clothing	range	l	il010230126	02/12/2017	02/12/2017	7.99	-	-
im01101047	game top	team shirt with emblem	clothing	range	m	il010230127	04/25/2017	04/25/2017	24.99	-	-
im01101048	premium bat	high quaity basball bat	equipment	-	-	il010230128	05/31/2017	05/31/2017	149	11/09/2023	11/09/2023
im01101048	premium bat	high quaity basball bat	equipment	-	-	il010230128	11/09/2023	11/09/2023	99.99	-	-

8 rows returned in 0.00 seconds Download

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 first 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.

```
SELECT ctr_number, c.first_name, c.last_name, c.phone_number, c.email, id, s.first_name, s.last_name, s.email
FROM customers c JOIN sales_representative s
ON (s.id = c.sre_id);
```

The screenshot shows the APEX SQL Workshop interface. The SQL command is as follows:

```
1 SELECT ctr_number, c.first_name, c.last_name, c.phone_number, c.email, id, s.first_name, s.last_name, s.email
2 FROM customers c JOIN sales_representatives s
3 ON (s.id = c.sre_id);
```

The results are displayed in a table with 9 columns: CTR_NUMBER, FIRST_NAME, LAST_NAME, PHONE_NUMBER, EMAIL, ID, FIRST_NAME, LAST_NAME, and EMAIL. The table contains 3 rows of data.

CTR_NUMBER	FIRST_NAME	LAST_NAME	PHONE_NUMBER	EMAIL	ID	FIRST_NAME	LAST_NAME	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

3 rows returned in 0.00 seconds [Download](#)

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

- 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.

```
SELECT ctr_number, c.first_name, c.last_name, c.phone_number, c.email, s.id, s.first_name, s.last_name, s.email,  
       t.name "Team Name"  
FROM customers c JOIN sales_representative s  
ON (s.id = c.sre_id)  
JOIN teams t  
ON (t.id = c.tem_id);
```

APEX App Builder SQL Workshop Team Development Gallery

Search

TAM JIA HAO databasejiahaotam

SQL Commands Schema WKSP_DATABASEJIAHAOTAM

Language SQL Rows 10 Clear Command Find Tables Save Run

```
1 SELECT ctr_number, c.first_name, c.last_name, c.phone_number, c.email, s.id, s.first_name, s.last_name, s.email, t.name "Team Name"  
2 FROM customers c JOIN sales_representatives s  
3 ON ( s.id = c.sre_id)  
4 JOIN teams t  
5 ON (t.id = c.tem_id);
```

Results Explain Describe Saved SQL History

CTR_NUMBER	FIRST_NAME	LAST_NAME	PHONE_NUMBER	EMAIL	ID	FIRST_NAME	LAST_NAME	EMAIL	Team Name
c00001	Robert	Thornberry	01234567898	bob.thornberry@heatmail.com	sr01	Charles	Raymond	chray@obl.com	Rockets
c00101	John	Doe	03216547808	unknown@here.com	sr01	Charles	Raymond	chray@obl.com	Celtics
c01986	Maria	Galant	01442736589	margal87@delphiview.com	sr03	Barry	Speed	bspeed@obl.com	Rovers

3 rows returned in 0.00 seconds Download

Part 5: Applying Additional Conditions to a Join

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

```
SELECT ctr_number, c.first_name, c.last_name, c.phone_number, c.email, s.id, s.first_name, s.last_name, s.email,  
       t.name "Team Name"  
FROM customers c JOIN sales_representative s  
ON (s.id = c.sre_id)  
JOIN teams t  
ON (t.id = c.tem_id);  
WHERE c.ctr_number = 'c00001';
```

APEX App Builder SQL Workshop Team Development Gallery

Search

TAM JIA HAO databasejiahaoam

SQL Commands Schema WKSP_DATABASEJIAHAOT/

Language SQL Rows 10 Clear Command Find Tables Save Run

```
1 SELECT ctr_number, c.first_name, c.last_name, c.phone_number, c.email, s.id, s.first_name, s.last_name, s.email, t.name "Team Name"  
2 FROM customers c JOIN sales_representatives s  
3 ON ( s.id = c.sre_id)  
4 JOIN teams t  
5 ON (t.id = c.tem_id)  
6 WHERE c.ctr_number = 'c00001';
```

Results Explain Describe Saved SQL History

CTR_NUMBER	FIRST_NAME	LAST_NAME	PHONE_NUMBER	EMAIL	ID	FIRST_NAME	LAST_NAME	EMAIL	Team Name
c00001	Robert	Thornberry	01234567898	bob.thornberry@heatmail.com	sr01	Charles	Raymond	chray@obl.com	Rockets

1 rows returned in 0.01 seconds Download

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

```
SELECT 'The cost of the ' || i.name || ' on this day was ' || p.price AS "Item Details"
FROM items i JOIN price_history p
ON ('12/12/2016' BETWEEN p.start_date AND p.end_date) AND (i.item_number = 'im01101045');
```

The screenshot shows the APEX SQL Workshop interface. At the top, there's a navigation bar with 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. A search bar and user information 'TAM JIA HAO databasejiahaotam' are on the right. Below the navigation bar, the 'SQL Commands' section is active, showing a query with three lines: a SELECT statement with a string concatenation for the alias 'Item Details', a FROM clause joining 'items' and 'price_history', and an ON clause with a date range and item number filter. The 'Results' tab is selected, displaying a single row with the value 'The cost of the under shirt on this day was 14.99' under the column header 'Item Details'. At the bottom, it states '1 rows returned in 0.00 seconds' with a 'Download' link.

```
1 SELECT 'The cost of the ' || i.name || ' on this day was ' || p.price AS "Item Details"
2 FROM items i JOIN price_history p
3 ON ('12/12/2016' BETWEEN p.start_date AND p.end_date) AND (i.item_number = 'im01101045');
```

Item Details
The cost of the under shirt on this day was 14.99

1 rows returned in 0.00 seconds [Download](#)

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 2: Joining Tables Using JOIN

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

Part 1 : Use a Self-Join to Join a Table to Itself (S6L9 Objective 2)

1. Write a query that will display who the supervisor is for each of the sales representatives. The information should be displayed in two columns, the first column will be the first name and last name of the sales representative and the second will be the first name and last name of the supervisor. The column aliases should be Rep and Supervisor.

```
SELECT r.first_name || ' ' || r.last_name AS "Rep", s.first_name || ' ' || s.last_name AS "Supervisor"
FROM sales_representatives r JOIN sales_representatives s
ON (r.supervisor_id = s.id);
```

The screenshot shows the APEX SQL Workshop interface. The SQL Commands pane contains the following query:

```
1 SELECT r.first_name || ' ' || r.last_name AS "Rep", s.first_name || ' ' || s.last_name AS "Supervisor"
2 FROM sales_representatives r JOIN sales_representatives s
3 ON (r.supervisor_id = s.id);
```

The Results pane shows the output of the query:

Rep	Supervisor
Charles Raymond	Charles Raymond
Victoria Wright	Charles Raymond
Barry Speed	Charles Raymond

3 rows returned in 0.01 seconds [Download](#)

Part 2 : Use OUTER joins (S6L9 Objective 3)

1. Write a query that will display all of the team and customer information even if there is no match with the table on the left (team).

```
SELECT *  
FROM teams t LEFT OUTER JOIN customers c  
ON (t.id = c.tem-id);
```

The screenshot shows the APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. A search bar and user profile 'TAM JIA HAO' are on the right. The 'SQL Commands' tab is active, showing a query with three lines: 'SELECT *', 'FROM teams t LEFT OUTER JOIN customers c', and 'ON (t.id = c.tem-id);'. Below the query editor, the 'Results' tab displays a table with 13 columns: ID, NAME, NUMBER_OF_PLAYERS, DISCOUNT, CTR_NUMBER, EMAIL, FIRST_NAME, LAST_NAME, PHONE_NUMBER, CURRENT_BALANCE, SRE_ID, TEM_ID, and LOYALTY_CARD_NUMBER. The table contains four rows of data. At the bottom, it states '4 rows returned in 0.01 seconds' and provides a 'Download' link.

ID	NAME	NUMBER_OF_PLAYERS	DISCOUNT	CTR_NUMBER	EMAIL	FIRST_NAME	LAST_NAME	PHONE_NUMBER	CURRENT_BALANCE	SRE_ID	TEM_ID	LOYALTY_CARD_NUMBER
t001	Rockets	25	10	c00001	bob.thornberry@heatmail.com	Robert	Thornberry	01234567898	150	sr01	t001	-
t002	Celtics	42	20	c00101	unknown@here.com	John	Doe	03216547808	987.5	sr01	t002	-
t003	Rovers	8	-	c01986	margal87@delphiview.com	Maria	Galant	01442736589	125.65	sr03	t003	-
t004	Jets	10	5	-	-	-	-	-	-	-	-	-

4 rows returned in 0.01 seconds [Download](#)

Part 3 : Generating a Cartesian Product (S6L9 Objective 4)

1. Create a Cartesian product between the customer and sales representative tables.

```
SELECT *  
FROM customers CROSS JOIN sales_representatives;
```

The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```
1 SELECT *  
2 FROM customers CROSS JOIN sales_representatives;
```

The results are displayed in a table with 15 columns: CTR_NUMBER, EMAIL, FIRST_NAME, LAST_NAME, PHONE_NUMBER, CURRENT_BALANCE, SRE_ID, TEM_ID, LOYALTY_CARD_NUMBER, ID, EMAIL, FIRST_NAME, LAST_NAME, PHONE_NUMBER, COMMISSION_RATE, and SUPERVISOR_ID. The table contains 10 rows of data, representing the Cartesian product of the customers and sales_representatives tables.

CTR_NUMBER	EMAIL	FIRST_NAME	LAST_NAME	PHONE_NUMBER	CURRENT_BALANCE	SRE_ID	TEM_ID	LOYALTY_CARD_NUMBER	ID	EMAIL	FIRST_NAME	LAST_NAME	PHONE_NUMBER	COMMISSION_RATE	SUPERVISOR_ID
c00001	bob.thornberry@heatmail.com	Robert	Thornberry	01234567898	150	sr01	1001	-	sr01	chray@obl.com	Charles	Raymond	0134598761	10	sr01
c00012	Jjones@freemail.com	Jennifer	Jones	01505214598	0	-	-	lc1015	sr01	chray@obl.com	Charles	Raymond	0134598761	10	sr01
c00101	unknown@here.com	John	Doe	03216547808	9875	sr01	1002	-	sr01	chray@obl.com	Charles	Raymond	0134598761	10	sr01
c00103	MurciaA@globaltech.com	Andrew	Murcia	0775246890	85	-	-	lc2341	sr01	chray@obl.com	Charles	Raymond	0134598761	10	sr01
c01986	margal87@delphiview.com	Maria	Galant	01442736589	125.65	sr03	1003	-	sr01	chray@obl.com	Charles	Raymond	0134598761	10	sr01
c02001	brianrog@hootech.com	Brian	Rogers	01654564898	50	-	-	lc4587	sr01	chray@obl.com	Charles	Raymond	0134598761	10	sr01
c00001	bob.thornberry@heatmail.com	Robert	Thornberry	01234567898	150	sr01	1001	-	sr02	vwright@obl.com	Victoria	Wright	0134598762	5	sr01
c00012	Jjones@freemail.com	Jennifer	Jones	01505214598	0	-	-	lc1015	sr02	vwright@obl.com	Victoria	Wright	0134598762	5	sr01
c00101	unknown@here.com	John	Doe	03216547808	9875	sr01	1002	-	sr02	vwright@obl.com	Victoria	Wright	0134598762	5	sr01
c00103	MurciaA@globaltech.com	Andrew	Murcia	0775246890	85	-	-	lc2341	sr02	vwright@obl.com	Victoria	Wright	0134598762	5	sr01

The screenshot shows the APEX SQL Workshop interface. The SQL command entered is:

```
1 SELECT *  
2 FROM customers CROSS JOIN sales_representatives;
```

The results are displayed in a table with 15 columns: CTR_NUMBER, EMAIL, FIRST_NAME, LAST_NAME, PHONE_NUMBER, CURRENT_BALANCE, SRE_ID, TEM_ID, LOYALTY_CARD_NUMBER, ID, EMAIL, FIRST_NAME, LAST_NAME, PHONE_NUMBER, COMMISSION_RATE, and SUPERVISOR_ID. The table contains 10 rows of data, representing the Cartesian product of the customers and sales_representatives tables.

CTR_NUMBER	EMAIL	FIRST_NAME	LAST_NAME	PHONE_NUMBER	CURRENT_BALANCE	SRE_ID	TEM_ID	LOYALTY_CARD_NUMBER	ID	EMAIL	FIRST_NAME	LAST_NAME	PHONE_NUMBER	COMMISSION_RATE	SUPERVISOR_ID
c00103	MurciaA@globaltech.com	Andrew	Murcia	0775246890	85	-	-	lc2341	sr01	chray@obl.com	Charles	Raymond	0134598761	10	sr01
c01986	margal87@delphiview.com	Maria	Galant	01442736589	125.65	sr03	1003	-	sr01	chray@obl.com	Charles	Raymond	0134598761	10	sr01
c02001	brianrog@hootech.com	Brian	Rogers	01654564898	50	-	-	lc4587	sr01	chray@obl.com	Charles	Raymond	0134598761	10	sr01
c00001	bob.thornberry@heatmail.com	Robert	Thornberry	01234567898	150	sr01	1001	-	sr02	vwright@obl.com	Victoria	Wright	0134598762	5	sr01
c00012	Jjones@freemail.com	Jennifer	Jones	01505214598	0	-	-	lc1015	sr02	vwright@obl.com	Victoria	Wright	0134598762	5	sr01
c00101	unknown@here.com	John	Doe	03216547808	9875	sr01	1002	-	sr02	vwright@obl.com	Victoria	Wright	0134598762	5	sr01
c00103	MurciaA@globaltech.com	Andrew	Murcia	0775246890	85	-	-	lc2341	sr02	vwright@obl.com	Victoria	Wright	0134598762	5	sr01

More than 10 rows available. Increase rows selector to view more rows.
10 rows returned in 0.00 seconds