



UTM
UNIVERSITI TEKNOLOGI MALAYSIA

DATA BASE (SECD2523-06)
LAB 4 : SQL 4-DML3

SECTION:	06
NAME:	Kwek Jia Cong (A22EC0122)
INSTRUCTORS:	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.

Part 1: Creating Natural Joins.

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

Part 2: Creating Joins with the USING Clause

1. Adapt the previous query answer to use the USING clause instead of a natural join.
2. Display all of the information about items and their price history by joining the items and price_history tables.

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.

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.

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.

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

Part 1: Creating Natural Joins.

- Display all of the information about sales representatives and their addresses using a natural join.
- `SELECT * FROM SALES_REPRESENTATIVES NATURAL JOIN SALES REP ADDRESSES;`

1 `SELECT * FROM SALES_REPRESENTATIVES NATURAL JOIN SALES REP ADDRESSES;`

Results										
ID	EMAIL	FIRST_NAME	LAST_NAME	PHONE_NUMBER	COMMISSION_RATE	SUPERVISOR_ID	ADDRESS_LINE_1	ADDRESS_LINE_2	CITY	ZIP_CODE
sr03	bspeed@obl.com	Barry	Speed	0134598763	5	sr01	12 Junction Row	Skinflats	Detroit	DT52564
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

3 rows returned in 0.03 seconds [Download](#)

- 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, email, phone_number, address_line_1, address_line_2, city FROM SALES_REPRESENTATIVES NATURAL JOIN SALES REP ADDRESSES;`

2 `SELECT id, first_name, last_name, email, phone_number, address_line_1, address_line_2, city FROM SALES_REPRESENTATIVES NATURAL JOIN SALES REP ADDRESSES;`

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

3 rows returned in 0.04 seconds [Download](#)

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, email, phone_number, address_line_1, address_line_2, city FROM SALES_REPRESENTATIVES NATURAL JOIN SALES REP ADDRESSES using (id);`

3 `SELECT id, first_name, last_name, email, phone_number, address_line_1, address_line_2, city FROM SALES_REPRESENTATIVES JOIN SALES REP ADDRESSES using(id);`

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

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

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(item_number);

4 SELECT * FROM ITEMS JOIN PRICE_HISTORY USING(item_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	-	-	
im01101046	socks	team socks with emblem	clothing	range	l	il010230126	02/12/2017	02/12/2017	.799	-	-	
im01101047	game top	team shirt with emblem	clothing	range	m	il010230127	04/25/2017	04/25/2017	24.99	-	-	
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	01/26/2017	01/26/2017	15.99	-	-	
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	
im01101048	premium bat	high qualy baseball bat	equipment	-	-	il010230128	05/31/2017	05/31/2017	149	11/09/2023	11/09/2023	
im01101048	premium bat	high qualy baseball bat	equipment	-	-	il010230128	11/09/2023	11/09/2023	99.99	-	-	

8 rows returned in 0.03 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 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.

- 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", c.sre_id AS "Sales_Rep ID",
 s.first_name AS "Sales_Rep First Name", s.last_name AS "Sales_Rep Last Name",
 s.email AS "Sales_Rep Email"
 FROM CUSTOMERS c JOIN SALES_REPRESENTATIVES s ON (c.sre_id = s.id);

1	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",							
2	c.email AS "Customer Email", c.sre_id AS "Sales_Rep ID", s.first_name AS "Sales_Rep First Name", s.last_name AS "Sales_Rep Last Name", s.email AS "Sales_Rep Email"							
3	FROM CUSTOMERS c JOIN SALES_REPRESENTATIVES s ON (c.sre_id = s.id);							
Results Explain Describe Saved SQL History								
Customer Number	Customer First Name	Customer Last Name	Customer Phone Number	Customer Email	Sales_Rep ID	Sales_Rep First Name	Sales_Rep Last Name	Sales_Rep Email
c00001	Robert	Thornberry	01234567898	bob.thornberry@heatmail.com	sr01	Charles	Raymond	chray@obl.com
c01986	Maria	Galant	01442736589	margal87@delphiview.com	sr03	Barry	Speed	bspeed@obl.com
c00101	John	Doe	03216547808	unknown@here.com	sr01	Charles	Raymond	chray@obl.com

3 rows returned in 0.01 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 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", c.sre_id AS "Sales_Rep ID",
s.first_name AS "Sales_Rep First Name", s.last_name AS "Sales_Rep Last Name",
s.email AS "Sales_Rep Email", t.name AS "Team Name"
FROM CUSTOMERS c JOIN SALES_REPRESENTATIVES s ON (c.sre_id = s.id);
JOIN TEAMS t ON (c.team_id = t.id);
```

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", c.sre_id AS "Sales_Rep ID", s.first_name AS "Sales_Rep First Name", s.last_name AS "Sales_Rep Last Name", s.email AS "Sales_Rep Email", t.name AS "Team Name" FROM CUSTOMERS c JOIN SALES_REPRESENTATIVES s ON (c.sre_id = s.id) JOIN TEAMS t ON (c.team_id = t.id);										
Results	Explain	Describe	Saved SQL	History						
Customer Number	Customer First Name	Customer Last Name	Customer Phone Number	Customer Email	Sales_Rep ID	Sales_Rep First Name	Sales_Rep Last Name	Sales_Rep Email	Team Name	
c00001	Robert	Thornberry	01234567898	bob.thornberry@heatmail.com	sr01	Charles	Raymond	chray@obl.com	Rockets	
c01986	Maria	Galant	01442736589	margal87@delphiview.com	sr03	Barry	Speed	bspeed@obl.com	Rovers	
c00101	John	Doe	03216547808	unknown@here.com	sr01	Charles	Raymond	chray@obl.com	Celtics	

3 rows returned in 0.04 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 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", c.sre_id AS "Sales_Rep ID",
s.first_name AS "Sales_Rep First Name", s.last_name AS "Sales_Rep Last Name",
s.email AS "Sales_Rep Email", t.name AS "Team Name"
FROM CUSTOMERS c JOIN SALES_REPRESENTATIVES s ON (c.sre_id = s.id);
JOIN TEAMS t ON (c.team_id = t.id) WHERE c.ctr_number = ('c00001');
```

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", c.sre_id AS "Sales_Rep ID", s.first_name AS "Sales_Rep First Name", s.last_name AS "Sales_Rep Last Name", s.email AS "Sales_Rep Email", t.name AS "Team Name" FROM CUSTOMERS c JOIN SALES_REPRESENTATIVES s ON (c.sre_id = s.id) JOIN TEAMS t ON (c.team_id = t.id) WHERE c.ctr_number = ('c00001');										
Results	Explain	Describe	Saved SQL	History						
Customer Number	Customer First Name	Customer Last Name	Customer Phone Number	Customer Email	Sales_Rep ID	Sales_Rep First Name	Sales_Rep Last Name	Sales_Rep 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 "Output"
FROM ITEMS i JOIN PRICE_HISTORY p ON (i.item_number = p.item_number)
WHERE i.item_number = ('im01101045') AND TO_DATE ('12-Dec-2016','DD-Mon-YYYY')
BETWEEN p.start_date AND p.end_date;

```
1 SELECT 'The cost of the'||i.name||' on this day was'||p.price AS "Output" FROM ITEMS i JOIN PRICE_HISTORY p ON (i.item_number = p.item_number) WHERE i.item_number = ('im01101045') AND TO_DATE('
2 p.end_date';

Results Explain Describe Saved SQL History
```

Output

```
The cost of the under shirt on this day was 14.99
1 rows returned in 0.03 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.

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).

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

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

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

- 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 rep.first_name || ' ' || rep.last_name AS "Rep", sup.first_name || ' ' || sup.last_name AS "Supervisor" FROM SALES_REPRESENTATIVES rep JOIN SALES_REPRESENTATIVES sup ON (rep.supervisor_id = sup.id);

1	SELECT rep.first_name ' ' rep.last_name AS "Rep" , sup.first_name ' ' sup.last_name AS "supervisor" FROM SALES_REPRESENTATIVES rep JOIN SALES_REPRESENTATIVES sup
2	ON (rep.supervisor_id = sup.id);
Results	
Explain Describe Saved SQL History	
Rep	
Barry Speed	Charles Raymond
Charles Raymond	Charles Raymond
Victoria Wright	Charles Raymond
3 rows returned in 0.03 seconds Download	

Part 2 : Use OUTER joins (S6L9 Objective 3)

- 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 CUSTOMER c ON (c.team_id = t.id);

3	SELECT * FROM TEAMS t LEFT OUTER JOIN CUSTOMERS c ON (c.team_id = t.id);											
Results												
Explain Describe Saved SQL History												
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	-
t003	Rovers	8	-	c01986	margal87@delphiview.com	Maria	Galant	01442736589	125.65	sr03	t003	-
t002	Celtics	42	20	c00101	unknown@here.com	John	Doe	03216547808	987.5	sr01	t002	-
t004	Jets	10	5	-	-	-	-	-	-	-	-	-
rows returned in 0.03 seconds Download												

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

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

- SELECT * FROM CUSTOMERS, SALES_REPRESENTATIVES ;

4	SELECT * FROM CUSTOMERS, SALES_REPRESENTATIVES;														
Results															
Explain Describe Saved SQL History															
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	sr03	bspeed@obl.com	Barry	Speed	0134598765	5	sr01
c00001	bob.thornberry@heatmail.com	Robert	Thornberry	01234567898	150	sr01	t001	-	sr03	bspeed@obl.com	Barry	Speed	0134598765	5	sr01
c00201	brianrog@hootech.com	Brian	Rogers	01654564898	50	-	-	lc4587	sr03	bspeed@obl.com	Barry	Speed	0134598765	5	sr01
c00102	jones@femail.com	Jennifer	Jones	01505214598	0	-	-	lc1015	sr03	bspeed@obl.com	Barry	Speed	0134598765	5	sr01
c01986	margal87@delphiview.com	Maria	Galant	01442736589	125.65	sr05	t003	-	sr03	bspeed@obl.com	Barry	Speed	0134598765	5	sr01
c00101	unknown@here.com	John	Doe	03216547808	987.5	sr01	t002	-	sr03	bspeed@obl.com	Barry	Speed	0134598765	5	sr01
c00103	MurciaA@globaltech.com	Andrew	Murcia	0775246890	85	-	-	lc2341	sr01	chray@obl.com	Charles	Raymond	0134598761	10	sr01
c00001	bob.thornberry@heatmail.com	Robert	Thornberry	01234567898	150	sr01	t001	-	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
c00012	jones@femail.com	Jennifer	Jones	01505214598	0	-	-	lc1015	sr01	chray@obl.com	Charles	Raymond	0134598761	10	sr01
More than 10 rows available. Increase rows selector to view more rows.															
10 rows returned in 0.02 seconds Download															