



MSIT 3860 – Term Project Final Deliverable

MSIT 3860 – Data Management for Info Tech

Term Project Final Deliverable

AMAZON CASE STUDY

Prexa Dedhia

Clark University

05/04/2020

Table of Contents

1	Structural Business Rule.....	3
2	Conceptual ERD.....	4
3	Logical ERD.....	5
4	SQL Addressing Five Aspects.....	6
5	Creation Of Index.....	16

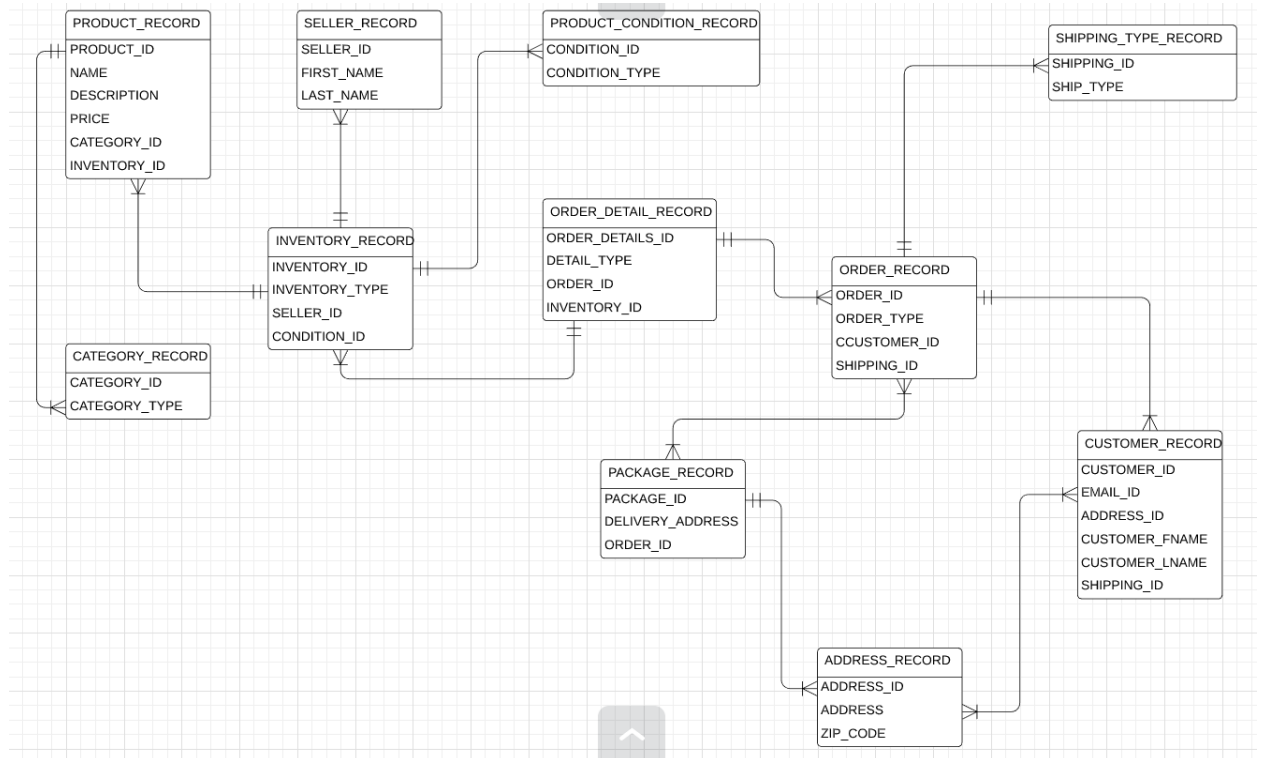


1. Structural Business Rules

- Inventory record has one and only one product record
- Inventory record has one and only one condition record
- Inventory record has one and only one seller record
- Inventory record has one or more order detail record
- Product record has one or more inventory record
- Product record has one and only one category record; category record has one or more product record
- Seller record has one or more inventory record
- Order detail record has one and only one inventory record
- Order detail record has one and only one order record
- Order record has one or more order detail record
- Order record has one and only one customer record
- Order record has one and only one package record
- Order record has one and only one shipping type record
- shipping type record has one or more order record
- package record has one and only one order record
- Customer record has one or more order record
- Customer record has one or more address record
- address record has one or more Customer record
- address record has one or more package record
- package record has one and only one address record

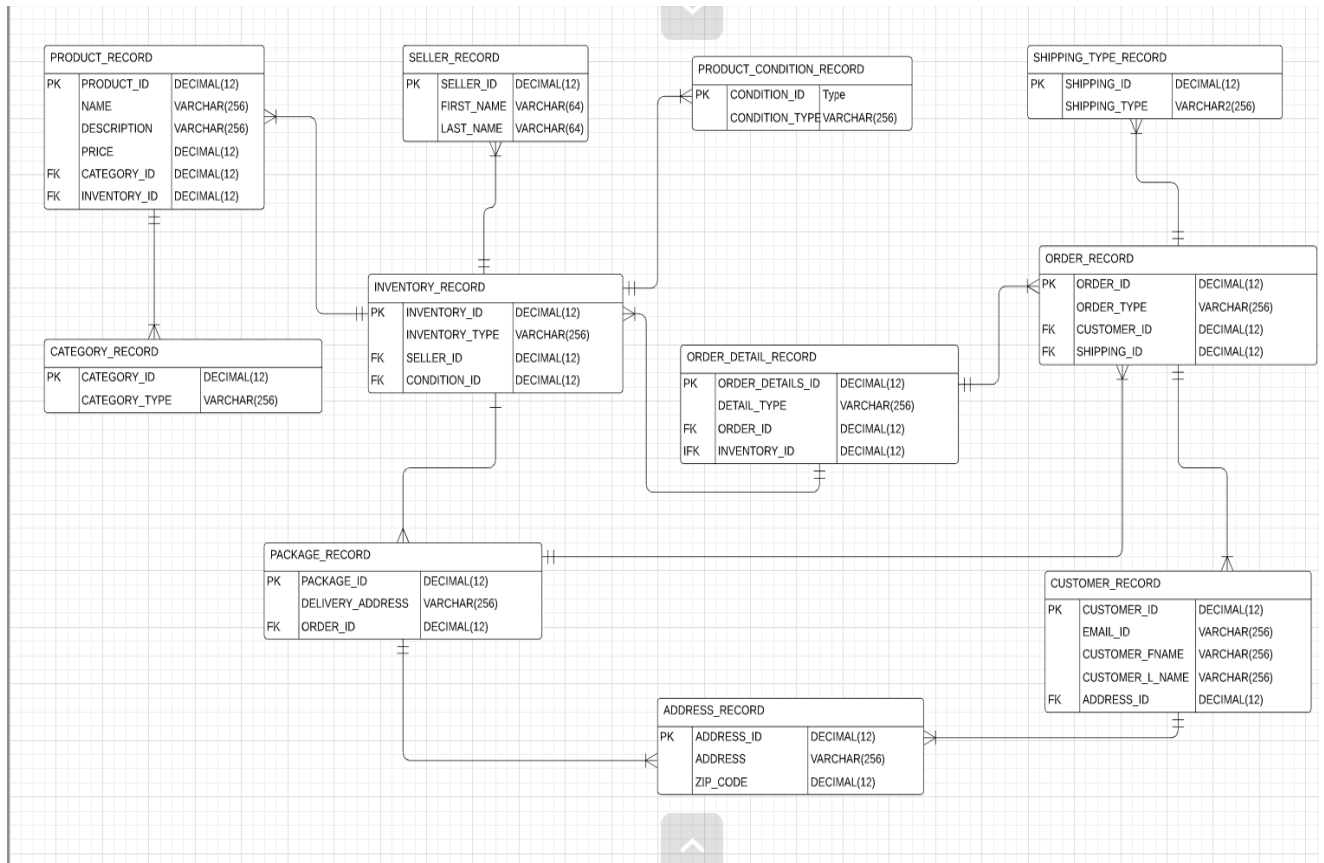


2. Conceptual ERD



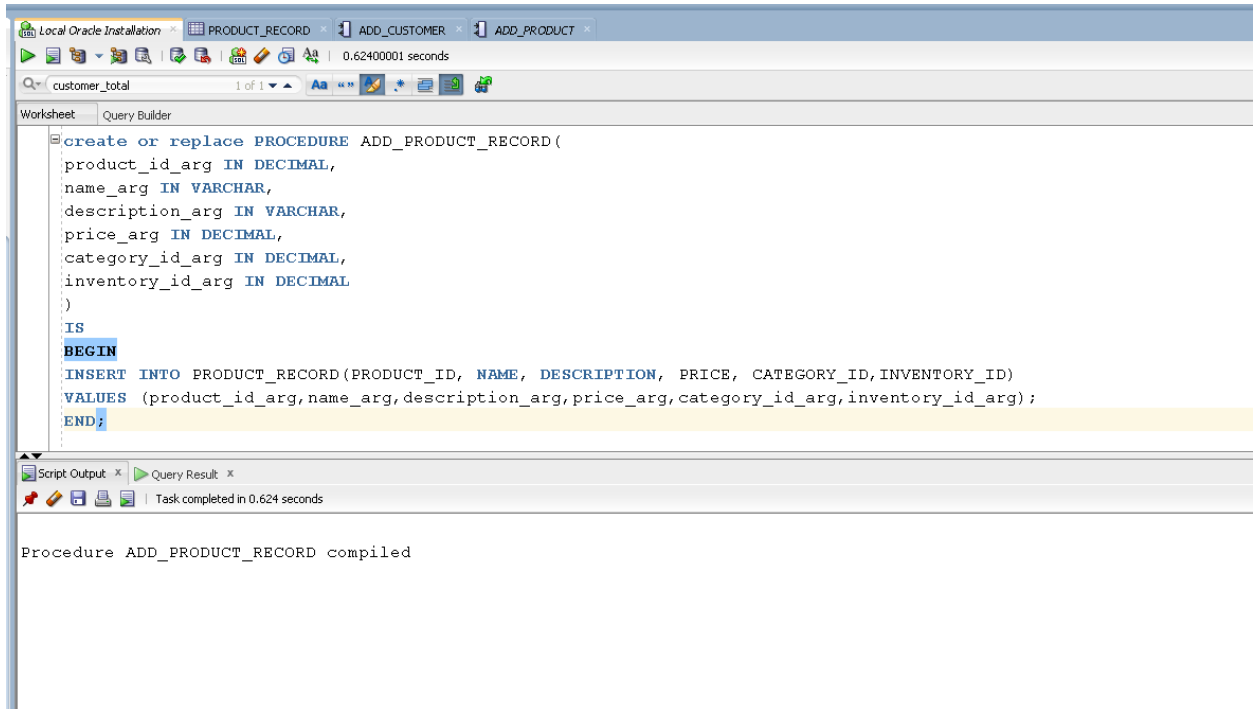


3. Logical ERD



4. Screenshots of the SQL addressing the five Aspect .

Aspect 1: New Product Created by Seller

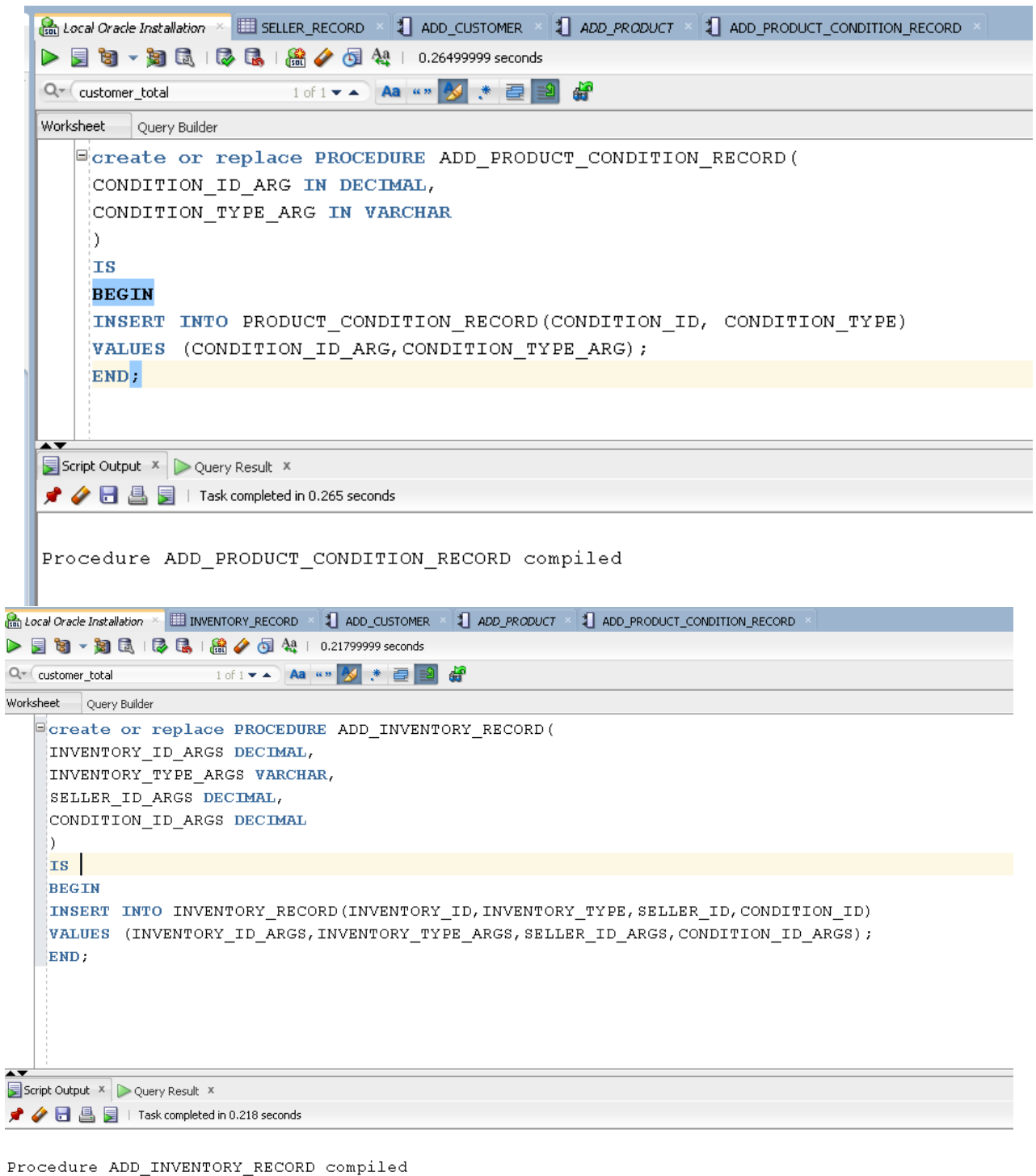


The screenshot shows the SQL Developer interface with the following details:

- Top Tab Bar:** Local Oracle Installation, PRODUCT_RECORD, ADD_CUSTOMER, ADD_PRODUCT.
- Toolbar:** Includes icons for running, saving, and other database actions. A timer shows 0.62400001 seconds.
- Search Bar:** Contains the text "customer_total".
- Worksheet Tab:** Labeled "Query Builder".
- SQL Editor:** Contains the following PL/SQL code:

```
create or replace PROCEDURE ADD_PRODUCT_RECORD (  
    product_id_arg IN DECIMAL,  
    name_arg IN VARCHAR,  
    description_arg IN VARCHAR,  
    price_arg IN DECIMAL,  
    category_id_arg IN DECIMAL,  
    inventory_id_arg IN DECIMAL  
)  
IS  
BEGIN  
    INSERT INTO PRODUCT_RECORD (PRODUCT_ID, NAME, DESCRIPTION, PRICE, CATEGORY_ID, INVENTORY_ID)  
    VALUES (product_id_arg, name_arg, description_arg, price_arg, category_id_arg, inventory_id_arg);  
END;
```
- Script Output Tab:** Shows the message "Task completed in 0.624 seconds".
- Query Result Tab:** Shows the message "Procedure ADD_PRODUCT_RECORD compiled".

MSIT 3860 – Term Project Final Deliverable



The image displays two screenshots of the Oracle SQL Developer interface, showing the successful compilation of two PL/SQL procedures.

Top Screenshot:

- Worksheet:** Contains the following PL/SQL code:

```
create or replace PROCEDURE ADD_PRODUCT_CONDITION_RECORD (  
    CONDITION_ID_ARG IN DECIMAL,  
    CONDITION_TYPE_ARG IN VARCHAR  
)  
IS  
BEGIN  
    INSERT INTO PRODUCT_CONDITION_RECORD (CONDITION_ID, CONDITION_TYPE)  
    VALUES (CONDITION_ID_ARG, CONDITION_TYPE_ARG);  
END;
```
- Script Output:** Displays the message: "Procedure ADD_PRODUCT_CONDITION_RECORD compiled".

Bottom Screenshot:

- Worksheet:** Contains the following PL/SQL code:

```
create or replace PROCEDURE ADD_INVENTORY_RECORD (  
    INVENTORY_ID_ARGS DECIMAL,  
    INVENTORY_TYPE_ARGS VARCHAR,  
    SELLER_ID_ARGS DECIMAL,  
    CONDITION_ID_ARGS DECIMAL  
)  
IS  
BEGIN  
    INSERT INTO INVENTORY_RECORD (INVENTORY_ID, INVENTORY_TYPE, SELLER_ID, CONDITION_ID)  
    VALUES (INVENTORY_ID_ARGS, INVENTORY_TYPE_ARGS, SELLER_ID_ARGS, CONDITION_ID_ARGS);  
END;
```
- Script Output:** Displays the message: "Procedure ADD_INVENTORY_RECORD compiled".



MSIT 3860 – Term Project Final Deliverable

Local Oracle Installation x ADD_CUSTOMER x ADD_PRODUCT x ADD_PRODUCT_CONDITION_RECORD x INVENTORY_RECORD x

0.109 seconds

customer_total 1 of 1

Worksheet Query Builder

```
BEGIN
  ADD_SELLER_RECORD(5, 'SHREY', 'SHAH');
  ADD_SELLER_RECORD(8, 'BILL', 'CLINTON');
  ADD_PRODUCT_CONDITION_RECORD(993, 'USED');
  ADD_PRODUCT_CONDITION_RECORD(984, 'NEW');
  ADD_INVENTORY_RECORD(0911, 'ELECTRONICS', 5, 984);
  ADD_CATEGORY_RECORD(073, 'ELECTRONICS');
  ADD_CATEGORY_RECORD(094, 'APPLIANCES');
  ADD_PRODUCT_RECORD(222, 'HOLOGRAPHIC KEYBOARD', 'EMITS A THREE-DIMENSIONAL PROJECTION OF A KEYBOARD AND RECOGNIZES VIRTUAL KEY PRESSES FROM THE TY
  ADD_PRODUCT_RECORD(227, 'SELF-DRIVING VIDEO CAMERA', 'AUTOMATICALLY FOLLOWS A SUBJECT THAT IS BEING RECORDED', 700, 073, 0911);
  ADD_PRODUCT_RECORD(555, 'BULB', 'MILTI-COLOR PARTY BULB', 15, 073, 0911);
  ADD_PRODUCT_RECORD(955, 'TORCH', '10 KMS COVERAGE FLASH LIGHT FOR TRREKKES', 28, 073, 0011);
END;
/
```

Script Output x Query Result x

Task completed in 0.109 seconds

PL/SQL procedure successfully completed.

Local Oracle Installation x ADD_CUSTOMER x ADD_PRODUCT x ADD_PRODUCT_CONDITION_RECORD x INVENTORY_RECORD x

customer_total 1 of 1

Worksheet Query Builder

```
SELECT * FROM
PRODUCT_RECORD, INVENTORY_RECORD WHERE
inventory_record.inventory_type='ELECTRONICS' AND
PRODUCT_RECORD.PRICE < 30;
```

Script Output x Query Result x

SQL All Rows Fetched: 2 in 0 seconds

	PRODUCT_ID	NAME	DESCRIPTION	PRICE	CATEGORY_ID	INVENTORY_ID	INVENTORY_ID_1	INVENTORY_TYPE	SELLER_ID	CONDITION_ID
1	555	BULB	MILTI-COLOR PARTY BULB	15	73	911	911	ELECTRONICS	5	984
2	955	TORCH	10 KMS COVERAGE FLASH LIGHT FOR TRREKKES	28	73	11	911	ELECTRONICS	5	984



Aspect 2: Amazon Receipt of Product from Seller

Local Oracle Installation x CATEGORY_RECORD x ADD_CUSTOMER x ADD_PRODUCT x

0.88999999 seconds

customer_total 1 of 1

Worksheet Query Builder

```
create or replace PROCEDURE ADD_SELLER_RECORD (
  SELLER_ID_ARG IN DECIMAL,
  FIRST_NAME_ARG IN VARCHAR,
  LAST_NAME_ARG IN VARCHAR
)
IS
BEGIN
  INSERT INTO SELLER_RECORD (SELLER_ID, FIRST_NAME, LAST_NAME)
  VALUES (SELLER_ID_ARG, FIRST_NAME_ARG, LAST_NAME_ARG);
END;
```

Script Output x Query Result x

Task completed in 0.89 seconds

Procedure ADD_SELLER_RECORD compiled

Local Oracle Installation x ADD_CUSTOMER x ADD_PRODUCT x ADD_PRODUCT_CONDITION_RECORD x INVENTORY_RECORD x

customer_total 1 of 1

Worksheet Query Builder

```
SELECT * FROM
INVENTORY_RECORD, PRODUCT_RECORD
WHERE inventory_record.seller_id=3 AND product_record.inventory_id=11;
```

Script Output x Query Result x

All Rows Fetched: 4 in 0 seconds

	INVENTORY_ID	INVENTORY_TYPE	SELLER_ID	CONDITION_ID	PRODUCT_ID	NAME	DESCRIPTION	PRICE	CATEGORY_ID	INVENTORY_ID_1
1	11	APPLIANCES	3	3	133	MACYS_BULLET_FOR_KITCHEN	FASTER THAN YOUR REGULAR BULLET	800	3	11
2	11	APPLIANCES	3	3	123	KELLYS_JUICER	FRESH JUICE ANYTIME	768	3	11
3	11	APPLIANCES	3	3	677	JADES_FAN	HIGH SPEED, CEILING FAN	266	3	11
4	11	APPLIANCES	3	3	955	TORCH	10 KMS COVERAGE FLASH LIGHT FOR TRREKKES	28	73	11



Aspect 3: New Consumer Account

The screenshot shows the Oracle SQL Developer interface. The top toolbar includes icons for running queries, saving, and other database functions. The main window displays the 'Query Builder' tab with the following SQL code:

```
create or replace PROCEDURE ADD_ADDRESS_RECORD (  
    ADDRESS_ID_ARGS IN DECIMAL,  
    ADDRESS_ARGS IN VARCHAR,  
    ZIP_CODE_ARGS IN DECIMAL  
)  
IS  
BEGIN  
    INSERT INTO ADDRESS_RECORD (ADDRESS_ID, ADDRESS , ZIP_CODE)  
    VALUES (ADDRESS_ID_ARGS, ADDRESS_ARGS , ZIP_CODE_ARGS) ;  
END;
```

Below the code editor, the 'Script Output' tab shows the message: 'Procedure ADD_ADDRESS_RECORD compiled'. The 'Query Result' tab is also visible but empty.

The screenshot shows the Oracle SQL Developer interface. The top toolbar includes icons for running queries, saving, and other database functions. The main window displays the 'Query Builder' tab with the following SQL code:

```
create or replace PROCEDURE ADD_CUSTOMER_RECORD (  
    CUSTOMER_ID_ARGS IN DECIMAL,  
    EMAIL_ID_ARGS IN VARCHAR,  
    CUSTOMER_FNAME_ARGS IN VARCHAR,  
    CUSTOMER_LNAME_ARGS IN VARCHAR,  
    ADDRESS_ID_ARGS IN VARCHAR  
)  
IS  
BEGIN  
    INSERT INTO CUSTOMER_RECORD (CUSTOMER_ID, EMAIL_ID, CUSTOMER_FNAME, CUSTOMER_LNAME, ADDRESS_ID)  
    VALUES (CUSTOMER_ID_ARGS, EMAIL_ID_ARGS, CUSTOMER_FNAME_ARGS, CUSTOMER_LNAME_ARGS, ADDRESS_ID_ARGS) ;  
END;
```

Below the code editor, the 'Script Output' tab shows the message: 'Procedure ADD_CUSTOMER_RECORD compiled'. The 'Query Result' tab is also visible but empty.



MSIT 3860 – Term Project Final Deliverable

The screenshot shows the Oracle SQL Developer interface. The top toolbar includes icons for Local Oracle Installation, ADD_CUSTOMER, ADD_PRODUCT, ADD_PRODUCT_CONDITION_RECORD, and ADDRESS_RECORD. The search bar contains 'customer_total' and '1 of 1'. The Worksheet tab is active, displaying a PL/SQL procedure:

```
BEGIN
ADD_CUSTOMER_RECORD (8, 'VAISHHHU@GMAIL.COM', 'SANYA', 'JAIN', 1);
ADD_CUSTOMER_RECORD (11, 'PREXADED@CLARKU.EDU', 'PREXA', 'DEDHIA', 1);
END;
```

The bottom pane shows the Script Output tab with the message: "PL/SQL procedure successfully completed." The task completion time is 0.125 seconds.

The screenshot shows the Oracle SQL Developer interface. The top toolbar includes icons for Local Oracle Installation, ADD_CUSTOMER, ADD_PRODUCT, ADD_PRODUCT_CONDITION_RECORD, and ADDRESS_RECORD. The search bar contains 'customer_total' and '1 of 1'. The Worksheet tab is active, displaying a SQL query:

```
SELECT * FROM
CUSTOMER_RECORD WHERE
CUSTOMER_LNAME = 'JAIN';
```

The bottom pane shows the Query Result tab with the message: "All Rows Fetched: 1 in 0 seconds". The result is displayed in a table:

CUSTOMER_ID	EMAIL_ID	CUSTOMER_FNAME	CUSTOMER_LNAME	ADDRESS_ID
1	8 VAISHHHU@GMAIL.COM	SANYA	JAIN	1



Aspect 4: Product Purchase by Consumer

The screenshot shows the Oracle SQL Developer interface. The top toolbar includes icons for Local Oracle Installation, ADD_CUSTOMER, ADD_PRODUCT, and ADD_PRODUCT_CONDITION_RECORD. The main window displays a PL/SQL procedure named 'customer_total' with the following code:

```
BEGIN  
ADD_ORDER_DETAIL_RECORD (66, 'FRAGILE', 9, 11);  
END;  
/
```

The bottom pane shows the 'Script Output' tab with the message: 'PL/SQL procedure successfully completed.' The execution time is 0.608 seconds.

The screenshot shows the Oracle SQL Developer interface. The top toolbar includes icons for Local Oracle Installation, ADD_CUSTOMER, ADD_PRODUCT, and ADD_PRODUCT_CONDITION_RECORD. The main window displays a SQL query named 'customer_total' with the following code:

```
SELECT A.ORDER_DETAILS_ID, B.INVENTORY_ID, B.INVENTORY_TYPE, B.SELLER_ID, B.CONDITION_ID  
FROM  
ORDER_DETAIL_RECORD A  
LEFT OUTER JOIN  
INVENTORY_RECORD B  
ON A.INVENTORY_ID = B.INVENTORY_ID;
```

The bottom pane shows the 'Query Result' tab with the message: 'All Rows Fetched: 2 in 0.015 seconds'. The results are displayed in a table with 5 columns: ORDER_DETAILS_ID, INVENTORY_ID, INVENTORY_TYPE, SELLER_ID, and CONDITION_ID.

ORDER_DETAILS_ID	INVENTORY_ID	INVENTORY_TYPE	SELLER_ID	CONDITION_ID
1	776	700 PAINTING	4	4
2	887	11 APPLIANCES	3	3



MSIT 3860 – Term Project Final Deliverable

Local Oracle Installation x ADD_CUSTOMER x ADD_PRODUCT x ADD_PRODUCT_CONDITION_RECORD x INVENTORY_RECORD x

customer_total 1 of 1

Worksheet Query Builder

```

SELECT D.CUSTOMER_FNAME, D.CUSTOMER_LNAME, D.ADDRESS_ID, Y.ADDRESS FROM
CUSTOMER_RECORD D
LEFT OUTER JOIN
(SELECT A.ORDER_DETAILS_ID, B.ORDER_ID, B.ORDER_TYPE, B.CUSTOMER_ID
FROM
ORDER_DETAIL_RECORD A
LEFT OUTER JOIN
ORDER_RECORD B
ON A.ORDER_ID = B.ORDER_ID) C
ON
D.CUSTOMER_ID = C.CUSTOMER_ID
LEFT OUTER JOIN
ADDRESS_RECORD Y
ON
D.ADDRESS_ID = Y.ADDRESS_ID
;

```

Script Output x Query Result x

SQL | All Rows Fetched: 6 in 0 seconds

	CUSTOMER_FNAME	CUSTOMER_LNAME	ADDRESS_ID	ADDRESS
1	sakshi	shah	1	MAIN STREET, MA
2	sakshi	shah	1	MAIN STREET, MA
3	sakshi	shah	1	MAIN STREET, MA
4	john	wick	2	CLARK UNIVERSITY, MAIN OFFICE
5	SANYA	JAIN	1	MAIN STREET, MA
6	PREXA	DEDHIA	1	MAIN STREET, MA



Aspect 5: Product Shipment by Amazon

Local Oracle Installation x ADD_CUSTOMER x ADD_PRODUCT x ADD_PRODUCT_CONDITION_RECORD x INVENTORY_RECORD x

customer_total 1 of 1

Worksheet Query Builder

orial

```
SELECT a.shipping_type , b.order_id, b.order_type, b.customer_id
from
ORDER_RECORD b
left outer join
SHIPPING_TYPE_RECORD a
on
b.shipping_id = a.shipping_id;
```

Script Output x Query... x

SQL | All Rows Fetched: 2 in 0 seconds

	SHIPPING_TYPE	ORDER_ID	ORDER_TYPE	CUSTOMER_ID
1	PRIME	9	MEDIUM	1
2	STANDARD	89	LARGE	1



MSIT 3860 – Term Project Final Deliverable

Local Oracle Installation x ADD_CUSTOMER x ADD_PRODUCT x ADD_PRODUCT_CONDITION_RECORD x INVENTORY_RECORD x

customer_total 1 of 1

Worksheet Query Builder

orial

```
SELECT a.shipping_type , b.order_id, b.order_type, b.customer_id
from
ORDER_RECORD b
left outer join
SHIPPING_TYPE_RECORD a
on
b.shipping_id = a.shipping_id;
```

Script Output x Query... x

SQL | All Rows Fetched: 2 in 0 seconds

	SHIPPING_TYPE	ORDER_ID	ORDER_TYPE	CUSTOMER_ID
1	PRIME	9	MEDIUM	1
2	STANDARD	89	LARGE	1



5. Creation Of INDEX

The database will have multiple customer records, and to do some analysis region wise, its best to filter address based on zip-codes. Hence indexing zip-codes will help save a lot of time and help process the data quicker.

The screenshot displays the SQL Developer interface. The top toolbar includes icons for file operations, editing, and execution. The main window is titled 'customer_total' and shows a 'Query Builder' tab. The SQL editor contains the following code:

```
CREATE INDEX ZIP_CODE_IDX  
ON ADDRESS_RECORD (ZIP_CODE) ;
```

Below the editor, the 'Script Output' tab shows the result of the query execution:

```
Index ZIP_CODE_IDX created.
```

The status bar at the bottom indicates 'Task completed in 0.125 seconds'.



MSIT 3860 – Term Project Final Deliverable

Indexing inventory type will help save a ton of time as it is associated with most of the important tables and it helps retrieve majority of information.

The screenshot displays the Oracle SQL Developer interface. At the top, there are tabs for 'Local Oracle Installation', 'ADD_CUSTOMER', 'ADD_PRODUCT', and 'ADD_PRODUCT_CONDITION_RECORD'. Below the tabs is a toolbar with various icons and a status bar indicating '0.156 seconds'. The main workspace is divided into two panes: 'Worksheet' and 'Query Builder'. The 'Worksheet' pane contains the following SQL script:

```
CREATE INDEX INVENTORY_TYPE_IDX  
ON INVENTORY_RECORD (INVENTORY_TYPE);
```

Below the main workspace, there is a 'Script Output' pane and a 'Query Result' pane. The 'Script Output' pane shows the message: 'Index INVENTORY_TYPE_IDX created.' The 'Query Result' pane is empty. The status bar at the bottom indicates 'Task completed in 0.156 seconds'.