|  |
| --- |
|  |
| DBST 652: Computer Product Inventory Database |
|  |

**Group 3**

Jacob Peterson

Frankesha Robinson

Brandon Russell

Ronald Savarese-Rea

Felix Sedegbe

April 10, 2016

|  |
| --- |
|  |

Contents

[Plan of Work 2](#_Toc448075354)

[Database: Computer Product Inventory Database 2](#_Toc448075355)

[Overview: 2](#_Toc448075356)

[Purpose and Objective: 2](#_Toc448075357)

[Technical Details: 2](#_Toc448075358)

[DDL and DML: 2](#_Toc448075359)

[Project Timeline by Deliverable 3](#_Toc448075360)

[Entity Relationship Diagram 3](#_Toc448075361)

[Data Definition Language – Command Code 5](#_Toc448075362)

[Output/Log for DDL 7](#_Toc448075363)

[Query Results from Catalog 10](#_Toc448075364)

[DML – Insert Command Code 12](#_Toc448075365)

[Object Output/Log 18](#_Toc448075366)

[SQL Queries (Code & Output) 31](#_Toc448075367)

[Simple Queries 31](#_Toc448075368)

[Advanced Queries 33](#_Toc448075369)

[Misc Queries 35](#_Toc448075370)

[Database Maintenance 40](#_Toc448075371)

[Index Code 40](#_Toc448075372)

[Stored Procedure Code 40](#_Toc448075373)

[Alter Command Code 41](#_Toc448075374)

[Explain Tool 41](#_Toc448075375)

[References 46](#_Toc448075376)

Computer Product Inventory Database

# Plan of Work

## Database: Computer Product Inventory Database

## Overview:

For any store to be profitable it must be able to keep track of the inventory coming in and going out.  This is especially important so that the store can make sure it has enough of each product stocked.  It is also critical for the store to analyze whether certain products sell well or not.  If the store inventory is lost or deployed inefficiently then this can lead to loss of money and time.  For this project we will create an inventory database for a computer parts store.

## Purpose and Objective:

This database will keep track of all of the store’s major computer based products.  This includes computer towers, monitors, peripherals, software and printers.  Users of this database will be able to see what products are currently in stock and in what quantities.  This database will track customer data associated with purchases.  For each product this database will include information about the hardware and software specification of the device.  Tailored reports will be a big part of this database.  Store management will be able to see a breakdown of how well certain products are selling based on the time since they were added to the store compared against the number of a product sold during a specific period of time.  Additionally, they will be able to target customers for upgrade products based on reports about the last time a customer bought a certain type of product.

## Technical Details:

Diagram Tool:ER Assistant

Database:Oracle 11g, using Virtual Desktop Applications (VDA)

Hardware and Software:UMUC (VDA) Intel(R) Xeon(R).  Operating system is Linux.

## DDL and DML:

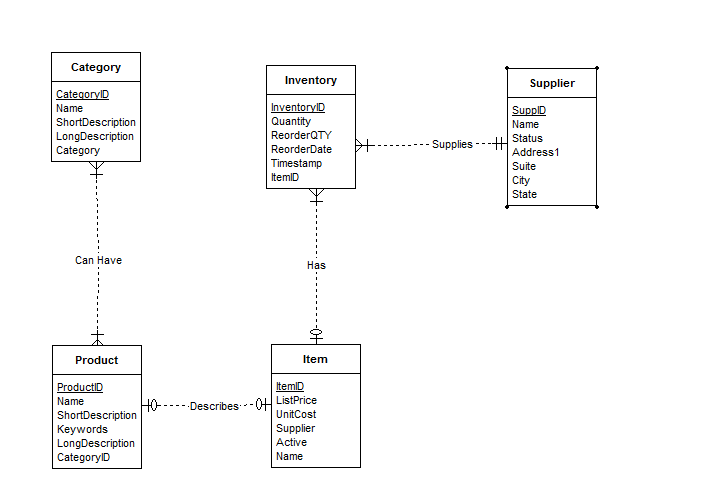
Data Definition Language (DDL) will be used to create the tables, keys, indexes and views needed for the database.  These commands will be used in the initial setup of the database.  Data Manipulation Language (DML) will be used to handle inserts, selects, updates, and filters with the where clause.  When a new asset is added to the database an insert command will be used.  If equipment details or location change an update command will be used.  Reports will be primarily generated using select statements paired with the where clause.  The primary language used will be Structured Query Language (SQL), and Oracle’s platform **specific PL/SQL.**

## Project Timeline by Deliverable

1. Develop Plan of Work: 2/1/216 – 2/8/16
2. Develop Entity Relationship Diagram: 2/9/16 – 2/16/16
3. Translate ERD into DDL and run DDL: 2/17/16 – 2/24/16
4. Generate and run 5 simple and 7 advanced SQL queries with documented results: 2/25/16 -3/1/16
5. Develop database maintenance scripts with output results: 3/2/16 – 3/9/16
6. Create stored procedures and test performance using the explain tool: 3/10/16 – 3/17/16
7. Document CPU execution times 3/18/16 – 3/25/16
8. Conduct object relational exercises and complete explain tool testing: 3/26/16 – 4/2/16
9. Compile report results: 4/3/16 – 4/10/16
10. Submit Lab: 4/10/16

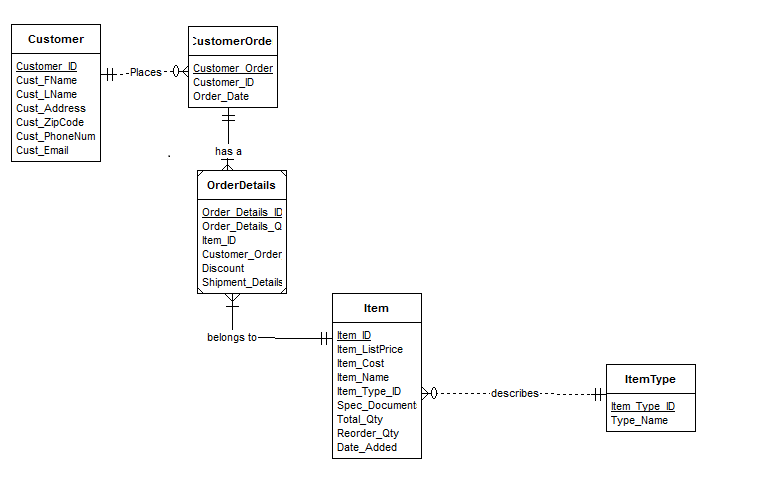
# Entity Relationship Diagram

Phase 1 ERD included many-to-many relationships between product and item. A subsequent ERD was generated to remove the anomaly and provide normal form recommendations.



**Resolved M-M relationships**

The final ERD below shows the optimized relationships between each entity. Each customer can have multiple orders. Each order can have multiple products



# Data Definition Language – Command Code

----SQL Script-----

SET echo on;

SET serveroutput on;

/\* Drop tables, sequence, and other objects you create\*/

DROP TABLE OrderDetails;

DROP TABLE Item;

DROP TABLE CustomerOrder;

DROP TABLE ItemType;

DROP TABLE Customer;

DROP SEQUENCE Customer\_Seq;

DROP SEQUENCE CustomerOrder\_Seq;

DROP SEQUENCE OrderDetails\_Seq;

DROP SEQUENCE Item\_Seq;

DROP SEQUENCE ItemType\_Seq;

/\* Create 5 tables \*/

CREATE TABLE Customer

(

Customer\_ID NUMBER NOT NULL,

Cust\_FName VARCHAR2 (50),

Cust\_LName VARCHAR2 (50),

Cust\_Address VARCHAR2 (100),

Cust\_ZipCode VARCHAR2(10),

Cust\_PhoneNum VARCHAR2(15),

Cust\_Email VARCHAR2(50),

CONSTRAINT pk\_customer PRIMARY KEY (Customer\_ID)

);

COMMENT ON TABLE Customer IS 'Table used to store Customer information';

DESCRIBE Customer;

CREATE TABLE CustomerOrder

(

Customer\_Order\_ID NUMBER NOT NULL,

Order\_Date DATE,

Customer\_ID NUMBER NOT NULL,

CONSTRAINT pk\_CustomerOrder PRIMARY KEY (Customer\_Order\_Id),

CONSTRAINT fk\_Customer FOREIGN KEY (Customer\_ID)

REFERENCES Customer

);

COMMENT ON TABLE CustomerOrder IS

'Table used to store main order information for each customer';

DESCRIBE CustomerOrder;

CREATE TABLE ItemType

(

Item\_Type\_ID NUMBER NOT NULL,

Type\_Name VARCHAR2(50),

CONSTRAINT pk\_ItemType PRIMARY KEY (Item\_Type\_ID)

);

COMMENT ON TABLE ItemType IS

'Table serves as a lookup or reference for the type of item an item is';

DESCRIBE ItemType;

CREATE TABLE Item

(

Item\_ID NUMBER NOT NULL,

Item\_ListPrice NUMBER,

Item\_Cost NUMBER,

Item\_Name VARCHAR2 (50),

Item\_Type\_ID NUMBER,

Spec\_Documents BLOB,

Total\_Qty NUMBER,

Reorder\_Qty NUMBER,

Date\_Added DATE,

CONSTRAINT pk\_Item PRIMARY KEY (Item\_ID),

CONSTRAINT fk\_ItemType FOREIGN KEY (Item\_Type\_ID)

REFERENCES ItemType

);

COMMENT ON TABLE Item IS

'Used to list all the necessary properties belonging to each specific item';

DESCRIBE Item;

CREATE TABLE OrderDetails

(

Order\_Details\_ID NUMBER NOT NULL,

Order\_Details\_Qty NUMBER,

Discount NUMBER,

Item\_ID NUMBER NOT NULL,

Customer\_Order\_ID NUMBER NOT NULL,

CONSTRAINT pk\_OrderDetails PRIMARY KEY (Order\_Details\_ID),

CONSTRAINT fk\_CustomerOrder FOREIGN KEY (Customer\_Order\_ID)

REFERENCES CustomerOrder,

CONSTRAINT fk\_OrderItem FOREIGN KEY (Item\_ID)

REFERENCES Item

);

COMMENT ON TABLE OrderDetails IS

'Contains the breakdown of each item ordered from CustomerOrder Table';

DESCRIBE OrderDetails;

/\* Create indexes on foreign keys\*/

CREATE INDEX fk\_Customer on CustomerOrder(Customer\_ID);

CREATE INDEX fk\_ItemType on Item(Item\_Type\_ID);

CREATE INDEX fk\_CustomerOrder on OrderDetails(Customer\_Order\_ID);

CREATE INDEX fk\_OrderItem on OrderDetails(Item\_ID);

/\* Create sequence\*/

CREATE SEQUENCE Customer\_Seq

START WITH 1

INCREMENT BY 1;

CREATE SEQUENCE CustomerOrder\_Seq

START WITH 1

INCREMENT BY 1;

CREATE SEQUENCE OrderDetails\_Seq

START WITH 1

INCREMENT BY 1;

CREATE SEQUENCE Item\_Seq

START WITH 1

INCREMENT BY 1;

CREATE SEQUENCE ItemType\_Seq

START WITH 1

INCREMENT BY 1;

/\* Data Dictionary query \*/

PURGE recyclebin;

/\*The next query returns a list of all objects in the user\_objects table\*/

SELECT /\*fixed\*/ object\_name, object\_type FROM user\_objects ORDER BY object\_type;

## Output/Log for DDL

SQL> SET serveroutput on

SQL> /\* Drop tables, sequence, and other objects you create\*/

SQL> DROP TABLE OrderDetails;

Table ORDERDETAILS dropped.

SQL> DROP TABLE Item;

Table ITEM dropped.

SQL> DROP TABLE CustomerOrder;

Table CUSTOMERORDER dropped.

SQL> DROP TABLE ItemType;

Table ITEMTYPE dropped.

SQL> DROP TABLE Customer;

Table CUSTOMER dropped.

SQL> DROP SEQUENCE Customer\_Seq;

Sequence CUSTOMER\_SEQ dropped.

SQL> DROP SEQUENCE CustomerOrder\_Seq;

Sequence CUSTOMERORDER\_SEQ dropped.

SQL> DROP SEQUENCE OrderDetails\_Seq;

Sequence ORDERDETAILS\_SEQ dropped.

SQL> DROP SEQUENCE Item\_Seq;

Sequence ITEM\_SEQ dropped.

SQL> DROP SEQUENCE ItemType\_Seq;

Sequence ITEMTYPE\_SEQ dropped.

SQL> /\* Create 5 tables \*/

SQL> CREATE TABLE Customer

(

Customer\_ID NUMBER NOT NULL,

Cust\_FName VARCHAR2 (50),

Cust\_LName VARCHAR2 (50),

Cust\_Address VARCHAR2 (100),

Cust\_ZipCode VARCHAR2(10),

Cust\_PhoneNum VARCHAR2(15),

Cust\_Email VARCHAR2(50),

CONSTRAINT pk\_customer PRIMARY KEY (Customer\_ID)

);

Table CUSTOMER created.

SQL> COMMENT ON TABLE Customer IS 'Table used to store Customer information';

Comment on table customer 'TABLE USED TO STORE CUSTOMER INFORMATION' succeeded.

SQL> DESCRIBE Customer;

SQL> CREATE TABLE CustomerOrder

(

Customer\_Order\_ID NUMBER NOT NULL,

Order\_Date DATE,

Customer\_ID NUMBER NOT NULL,

CONSTRAINT pk\_CustomerOrder PRIMARY KEY (Customer\_Order\_Id),

CONSTRAINT fk\_Customer FOREIGN KEY (Customer\_ID)

REFERENCES Customer

);

Table CUSTOMERORDER created.

SQL> COMMENT ON TABLE CustomerOrder IS

'Table used to store main order information for each customer';

Comment on table customerorder 'TABLE USED TO STORE MAIN ORDER INFORMATION FOR EACH CUSTOMER' succeeded.

SQL> DESCRIBE CustomerOrder;

SQL> CREATE TABLE ItemType

(

Item\_Type\_ID NUMBER NOT NULL,

Type\_Name VARCHAR2(50),

CONSTRAINT pk\_ItemType PRIMARY KEY (Item\_Type\_ID)

);

Table ITEMTYPE created.

SQL> COMMENT ON TABLE ItemType IS

'Table serves as a lookup or reference for the type of item an item is';

Comment on table itemtype 'TABLE SERVES AS A LOOKUP OR REFERENCE FOR THE TYPE OF ITEM AN ITEM IS' succeeded.

SQL> DESCRIBE ItemType;

SQL> CREATE TABLE Item

(

Item\_ID NUMBER NOT NULL,

Item\_ListPrice NUMBER,

Item\_Cost NUMBER,

Item\_Name VARCHAR2 (50),

Item\_Type\_ID NUMBER,

Spec\_Documents BLOB,

Total\_Qty NUMBER,

Reorder\_Qty NUMBER,

Date\_Added DATE,

CONSTRAINT pk\_Item PRIMARY KEY (Item\_ID),

CONSTRAINT fk\_ItemType FOREIGN KEY (Item\_Type\_ID)

REFERENCES ItemType

);

Table ITEM created.

SQL> COMMENT ON TABLE Item IS

'Used to list all the necessary properties belonging to each specific item';

Comment on table item 'USED TO LIST ALL THE NECESSARY PROPERTIES BELONGING TO EACH SPECIFIC ITEM' succeeded.

SQL> DESCRIBE Item;

SQL> CREATE TABLE OrderDetails

(

Order\_Details\_ID NUMBER NOT NULL,

Order\_Details\_Qty NUMBER,

Discount NUMBER,

Item\_ID NUMBER NOT NULL,

Customer\_Order\_ID NUMBER NOT NULL,

CONSTRAINT pk\_OrderDetails PRIMARY KEY (Order\_Details\_ID),

CONSTRAINT fk\_CustomerOrder FOREIGN KEY (Customer\_Order\_ID)

REFERENCES CustomerOrder,

CONSTRAINT fk\_OrderItem FOREIGN KEY (Item\_ID)

REFERENCES Item

);

Table ORDERDETAILS created.

SQL> COMMENT ON TABLE OrderDetails IS

'Contains the breakdown of each item ordered from CustomerOrder Table';

Comment on table orderdetails 'CONTAINS THE BREAKDOWN OF EACH ITEM ORDERED FROM CUSTOMERORDER TABLE' succeeded.

SQL> DESCRIBE OrderDetails;

Index FK\_CUSTOMER created.

SQL> CREATE INDEX fk\_ItemType on Item(Item\_Type\_ID);

Index FK\_ITEMTYPE created.

SQL> CREATE INDEX fk\_CustomerOrder on OrderDetails(Customer\_Order\_ID);

Index FK\_CUSTOMERORDER created.

SQL> CREATE INDEX fk\_OrderItem on OrderDetails(Item\_ID);

Index FK\_ORDERITEM created.

SQL> /\* Create sequence\*/

SQL> CREATE SEQUENCE Customer\_Seq

START WITH 1

INCREMENT BY 1;

Sequence CUSTOMER\_SEQ created.

SQL> CREATE SEQUENCE CustomerOrder\_Seq

START WITH 1

INCREMENT BY 1;

Sequence CUSTOMERORDER\_SEQ created.

SQL> CREATE SEQUENCE OrderDetails\_Seq

START WITH 1

INCREMENT BY 1;

Sequence ORDERDETAILS\_SEQ created.

SQL> CREATE SEQUENCE Item\_Seq

START WITH 1

INCREMENT BY 1;

Sequence ITEM\_SEQ created.

SQL> CREATE SEQUENCE ItemType\_Seq

START WITH 1

INCREMENT BY 1;

Sequence ITEMTYPE\_SEQ created.

SQL> /\* Data Dictionary query \*/

SQL> PURGE recyclebin;

PURGE recyclebin

SQL> /\*The next query returns a list of all objects in the user\_objects table\*/

SQL> SELECT /\*fixed\*/ object\_name, object\_type FROM user\_objects ORDER BY object\_type;

## Query Results from Catalog

"OBJECT\_NAME" "OBJECT\_TYPE"

"CUSTOM\_AUTH" "FUNCTION"

"CUSTOM\_HASH" "FUNCTION"

"APEX$\_WS\_LINKS\_PK" "INDEX"

"APEX$\_WS\_TAGS\_PK" "INDEX"

"APEX$\_WS\_FILES\_PK" "INDEX"

"APEX$\_ACL\_IDX1" "INDEX"

"APEX$\_WS\_ROWS\_IDX" "INDEX"

"APEX$\_WS\_HISTORY\_IDX" "INDEX"

"APEX$\_WS\_NOTES\_IDX1" "INDEX"

"APEX$\_WS\_NOTES\_IDX2" "INDEX"

"APEX$\_WS\_LINKS\_IDX1" "INDEX"

"APEX$\_WS\_LINKS\_IDX2" "INDEX"

"APEX$\_WS\_TAGS\_IDX1" "INDEX"

"APEX$\_WS\_TAGS\_IDX2" "INDEX"

"APEX$\_WS\_FILES\_IDX1" "INDEX"

"APEX$\_WS\_FILES\_IDX2" "INDEX"

"APEX$\_WS\_WEBPG\_SECHIST\_IDX1" "INDEX"

"SYS\_C007086" "INDEX"

"SYS\_C007087" "INDEX"

"SYS\_C006997" "INDEX"

"SYS\_C006999" "INDEX"

"DEMO\_USERS\_PK" "INDEX"

"DEMO\_CUSTOMERS\_PK" "INDEX"

"DEMO\_CUST\_NAME\_IX" "INDEX"

"DEMO\_ORDER\_PK" "INDEX"

"DEMO\_ORD\_CUSTOMER\_IX" "INDEX"

"DEMO\_PRODUCT\_INFO\_PK" "INDEX"

"DEMO\_ORDER\_ITEMS\_PK" "INDEX"

"APEX$\_ACL\_PK" "INDEX"

"APEX$\_WS\_WEBPG\_SECTIONS\_PK" "INDEX"

"APEX$\_WS\_ROWS\_PK" "INDEX"

"APEX$\_WS\_NOTES\_PK" "INDEX"

"PK\_SECTION" "INDEX"

"PK\_INSTR" "INDEX"

"PK\_CUSTOMER" "INDEX"

"PK\_COURSE" "INDEX"

"PK\_SCHEDULE" "INDEX"

"PK\_CLASS" "INDEX"

"FK\_COURSE\_LIST" "INDEX"

"FK\_INSTR" "INDEX"

"FK\_INSTRCLASS" "INDEX"

"FK\_CLASSSCHED" "INDEX"

"FK\_COURSECLASS" "INDEX"

"PK\_CUSTOMERORDER" "INDEX"

"PK\_ITEMTYPE" "INDEX"

"PK\_ITEM" "INDEX"

"PK\_ORDERDETAILS" "INDEX"

"FK\_CUSTOMER" "INDEX"

"FK\_ITEMTYPE" "INDEX"

"FK\_CUSTOMERORDER" "INDEX"

"FK\_ORDERITEM" "INDEX"

"SYS\_LOB0000020119C00010$$" "LOB"

"SYS\_LOB0000020123C00009$$" "LOB"

"SYS\_LOB0000020123C00008$$" "LOB"

"SYS\_LOB0000020083C00007$$" "LOB"

"SYS\_LOB0000020353C00006$$" "LOB"

"SYS\_LOB0000020098C00007$$" "LOB"

"SYS\_LOB0000020104C00166$$" "LOB"

"SYS\_LOB0000020104C00165$$" "LOB"

"SYS\_LOB0000020111C00007$$" "LOB"

"SYS\_LOB0000020098C00008$$" "LOB"

"DEMO\_CUST\_SEQ" "SEQUENCE"

"DEMO\_ORDER\_ITEMS\_SEQ" "SEQUENCE"

"DEMO\_ORD\_SEQ" "SEQUENCE"

"DEMO\_PROD\_SEQ" "SEQUENCE"

"DEMO\_USERS\_SEQ" "SEQUENCE"

"SECTIONNUM\_SEQ" "SEQUENCE"

"ITEMTYPE\_SEQ" "SEQUENCE"

"COURSENUM\_SEQ" "SEQUENCE"

"SCHEDNUM\_SEQ" "SEQUENCE"

"CUSTOMER\_SEQ" "SEQUENCE"

"CUSTOMERORDER\_SEQ" "SEQUENCE"

"ORDERDETAILS\_SEQ" "SEQUENCE"

"ITEM\_SEQ" "SEQUENCE"

"INSTRNUM\_SEQ" "SEQUENCE"

"APEX$\_WS\_TAGS" "TABLE"

"ORDERDETAILS" "TABLE"

"APEX$\_WS\_WEBPG\_SECTION\_HISTORY""TABLE"

"APEX$\_WS\_FILES" "TABLE"

"ITEM" "TABLE"

"ITEMTYPE" "TABLE"

"CUSTOMERORDER" "TABLE"

"INSTR\_CLASSES" "TABLE"

"CLASS\_SCHED" "TABLE"

"COURSE\_LIST" "TABLE"

"INSTR\_LIST" "TABLE"

"SECTION\_INFO" "TABLE"

"CUSTOMER" "TABLE"

"APEX$\_WS\_LINKS" "TABLE"

"APEX$\_WS\_NOTES" "TABLE"

"APEX$\_WS\_HISTORY" "TABLE"

"APEX$\_WS\_ROWS" "TABLE"

"APEX$\_WS\_WEBPG\_SECTIONS" "TABLE"

"APEX$\_ACL" "TABLE"

"DEMO\_STATES" "TABLE"

"DEMO\_ORDER\_ITEMS" "TABLE"

"DEMO\_PRODUCT\_INFO" "TABLE"

"INSTR\_OBJ\_TABLE" "TABLE"

"DEPT" "TABLE"

"EMP" "TABLE"

"DEMO\_USERS" "TABLE"

"DEMO\_CUSTOMERS" "TABLE"

"DEMO\_ORDERS" "TABLE"

"INSERT\_DEMO\_CUST" "TRIGGER"

"APEX$\_ACL\_T1" "TRIGGER"

"DEMO\_USERS\_T1" "TRIGGER"

"BI\_DEMO\_USERS" "TRIGGER"

"SCHEDCLASS\_TRIGGER" "TRIGGER"

"INSERT\_DEMO\_PROD" "TRIGGER"

"DEMO\_ORDER\_ITEMS\_GET\_PRICE" "TRIGGER"

"UPDATE\_ORDER\_TOTAL" "TRIGGER"

"INSERT\_DEMO\_ORDER\_ITEMS" "TRIGGER"

"APEX$\_WS\_WEBPG\_SECTIONS\_T1" "TRIGGER"

"APEX$\_WS\_ROWS\_T1" "TRIGGER"

"APEX$\_WS\_NOTES\_T1" "TRIGGER"

"APEX$\_WS\_FILES\_T1" "TRIGGER"

"APEX$\_WS\_TAGS\_T1" "TRIGGER"

"APEX$\_WS\_LINKS\_T1" "TRIGGER"

"PERSON\_TYP" "TYPE"

"INSTRUCTOR\_TYP" "TYPE"

"PERSON\_TYP" "TYPE BODY"

"INSTRUCTOR\_TYP" "TYPE BODY"

"SECTION\_INSTRUCTORS\_VIEW" "VIEW"

"INSTR\_TEACHING\_VIEW" "VIEW"

124 rows selected

# DML – Insert Command Code

/\* Insert data into each table \*/

INSERT INTO Customer (Customer\_ID, Cust\_FName, Cust\_LName, Cust\_Address, Cust\_ZipCode, Cust\_PhoneNum, Cust\_Email)

VALUES (Customer\_Seq.NEXTVAL, 'Bob', 'Roberts', '123 ABC Street','31088', '123-456-0000', 'BobRoberts@gmail.com');

INSERT INTO Customer (Customer\_ID, Cust\_FName, Cust\_LName, Cust\_Address, Cust\_ZipCode, Cust\_PhoneNum, Cust\_Email)

VALUES (Customer\_Seq.NEXTVAL, 'Mary', 'Jane', '45 Alberta Lane ','31771', '476-897-1111', 'MaryJane@gmail.com');

INSERT INTO Customer (Customer\_ID, Cust\_FName, Cust\_LName, Cust\_Address, Cust\_ZipCode, Cust\_PhoneNum, Cust\_Email)

VALUES (Customer\_Seq.NEXTVAL, 'Brandon', 'Russell', '123 Finegand Place ','31771', '234-123-222', 'HighFive34@gmail.com');

INSERT INTO Customer (Customer\_ID, Cust\_FName, Cust\_LName, Cust\_Address, Cust\_ZipCode, Cust\_PhoneNum, Cust\_Email)

VALUES (Customer\_Seq.NEXTVAL, 'James', 'Doe', '46 Alberta Lane ','31771', '476-897-1112', 'JamesDoe@gmail.com');

INSERT INTO Customer (Customer\_ID, Cust\_FName, Cust\_LName, Cust\_Address, Cust\_ZipCode, Cust\_PhoneNum, Cust\_Email)

VALUES (Customer\_Seq.NEXTVAL, 'Jim', 'Bob', '47 Alberta Lane ','31771', '476-897-1113', 'JimBob@gmail.com');

INSERT INTO Customer (Customer\_ID, Cust\_FName, Cust\_LName, Cust\_Address, Cust\_ZipCode, Cust\_PhoneNum, Cust\_Email)

VALUES (Customer\_Seq.NEXTVAL, 'Patricia', 'Russell', '48 Alberta Lane ','31771', '476-897-1114', 'PRussell@gmail.com');

INSERT INTO Customer (Customer\_ID, Cust\_FName, Cust\_LName, Cust\_Address, Cust\_ZipCode, Cust\_PhoneNum, Cust\_Email)

VALUES (Customer\_Seq.NEXTVAL, 'Jane', 'Johnson', '49 Alberta Lane ','31771', '476-897-1115', 'JJ123@gmail.com');

INSERT INTO Customer (Customer\_ID, Cust\_FName, Cust\_LName, Cust\_Address, Cust\_ZipCode, Cust\_PhoneNum, Cust\_Email)

VALUES (Customer\_Seq.NEXTVAL, 'Hyon', 'Yi', '50 Alberta Lane ','31771', '476-897-1116', 'HyonYi@gmail.com');

INSERT INTO Customer (Customer\_ID, Cust\_FName, Cust\_LName, Cust\_Address, Cust\_ZipCode, Cust\_PhoneNum, Cust\_Email)

VALUES (Customer\_Seq.NEXTVAL, 'Keith', 'Jordan', '51 Alberta Lane ','31771', '476-897-1117', 'KJ88@gmail.com');

INSERT INTO Customer (Customer\_ID, Cust\_FName, Cust\_LName, Cust\_Address, Cust\_ZipCode, Cust\_PhoneNum, Cust\_Email)

VALUES (Customer\_Seq.NEXTVAL, 'Pete', 'Smith', '52 Alberta Lane ','31771', '476-897-1118', 'PeteS33@gmail.com');

INSERT INTO Customer (Customer\_ID, Cust\_FName, Cust\_LName, Cust\_Address, Cust\_ZipCode, Cust\_PhoneNum, Cust\_Email)

VALUES (Customer\_Seq.NEXTVAL, 'Bobby', 'Robert', '124 ABC Street','31088', '223-456-0000', 'BobRobert@gmail.com');

INSERT INTO Customer (Customer\_ID, Cust\_FName, Cust\_LName, Cust\_Address, Cust\_ZipCode, Cust\_PhoneNum, Cust\_Email)

VALUES (Customer\_Seq.NEXTVAL, 'MaryJane', 'Joy', '46 Alberta Lane ','31771', '576-897-1111', 'MaryJane1@gmail.com');

INSERT INTO Customer (Customer\_ID, Cust\_FName, Cust\_LName, Cust\_Address, Cust\_ZipCode, Cust\_PhoneNum, Cust\_Email)

VALUES (Customer\_Seq.NEXTVAL, 'Brando', 'Russell', '124 Finegand Place ','31771', '334-123-222', 'H2ighFive34@gmail.com');

INSERT INTO Customer (Customer\_ID, Cust\_FName, Cust\_LName, Cust\_Address, Cust\_ZipCode, Cust\_PhoneNum, Cust\_Email)

VALUES (Customer\_Seq.NEXTVAL, 'Jay', 'Doe', '411 Alberta Lane ','31771', '471-897-1112', 'JDoe@gmail.com');

INSERT INTO Customer (Customer\_ID, Cust\_FName, Cust\_LName, Cust\_Address, Cust\_ZipCode, Cust\_PhoneNum, Cust\_Email)

VALUES (Customer\_Seq.NEXTVAL, 'Jimmie', 'Horton', '470 Alberta Lane ','31771', '476-897-1119', 'JimHorton@gmail.com');

INSERT INTO Customer (Customer\_ID, Cust\_FName, Cust\_LName, Cust\_Address, Cust\_ZipCode, Cust\_PhoneNum, Cust\_Email)

VALUES (Customer\_Seq.NEXTVAL, 'Patrick', 'Russell', '480 Alberta Lane ','31771', '476-897-1124', 'PRussell1@gmail.com');

INSERT INTO Customer (Customer\_ID, Cust\_FName, Cust\_LName, Cust\_Address, Cust\_ZipCode, Cust\_PhoneNum, Cust\_Email)

VALUES (Customer\_Seq.NEXTVAL, 'John', 'Johnson', '490 Alberta Lane ','31771', '476-897-1315', 'JJ124@gmail.com');

INSERT INTO Customer (Customer\_ID, Cust\_FName, Cust\_LName, Cust\_Address, Cust\_ZipCode, Cust\_PhoneNum, Cust\_Email)

VALUES (Customer\_Seq.NEXTVAL, 'Hyon', 'Yie', '500 Alberta Lane ','31771', '476-897-1516', 'HyonYi1@gmail.com');

INSERT INTO Customer (Customer\_ID, Cust\_FName, Cust\_LName, Cust\_Address, Cust\_ZipCode, Cust\_PhoneNum, Cust\_Email)

VALUES (Customer\_Seq.NEXTVAL, 'Mike', 'Jordan', '510 Alberta Lane ','31771', '476-997-1117', 'MJ88@gmail.com');

INSERT INTO Customer (Customer\_ID, Cust\_FName, Cust\_LName, Cust\_Address, Cust\_ZipCode, Cust\_PhoneNum, Cust\_Email)

VALUES (Customer\_Seq.NEXTVAL, 'Peter', 'Smith', '520 Alberta Lane ','31771', '476-899-1118', 'PeterS33@gmail.com');

INSERT INTO CustomerOrder (Customer\_Order\_ID, Customer\_ID, Order\_Date)

VALUES (CustomerOrder\_Seq.NEXTVAL, 1, '12-FEB-15');

INSERT INTO CustomerOrder (Customer\_Order\_ID, Customer\_ID, Order\_Date)

VALUES (CustomerOrder\_Seq.NEXTVAL, 2, '15-FEB-16');

INSERT INTO CustomerOrder (Customer\_Order\_ID, Customer\_ID, Order\_Date)

VALUES (CustomerOrder\_Seq.NEXTVAL, 3, '16-FEB-16');

INSERT INTO CustomerOrder (Customer\_Order\_ID, Customer\_ID, Order\_Date)

VALUES (CustomerOrder\_Seq.NEXTVAL, 4, '15-JAN-16');

INSERT INTO CustomerOrder (Customer\_Order\_ID, Customer\_ID, Order\_Date)

VALUES (CustomerOrder\_Seq.NEXTVAL, 5, '10-JAN-16');

INSERT INTO CustomerOrder (Customer\_Order\_ID, Customer\_ID, Order\_Date)

VALUES (CustomerOrder\_Seq.NEXTVAL, 6, '01-FEB-16');

INSERT INTO CustomerOrder (Customer\_Order\_ID, Customer\_ID, Order\_Date)

VALUES (CustomerOrder\_Seq.NEXTVAL, 7, '03-DEC-15');

INSERT INTO CustomerOrder (Customer\_Order\_ID, Customer\_ID, Order\_Date)

VALUES (CustomerOrder\_Seq.NEXTVAL, 8, '20-NOV-15');

INSERT INTO CustomerOrder (Customer\_Order\_ID, Customer\_ID, Order\_Date)

VALUES (CustomerOrder\_Seq.NEXTVAL, 9, '22-OCT-15');

INSERT INTO CustomerOrder (Customer\_Order\_ID, Customer\_ID, Order\_Date)

VALUES (CustomerOrder\_Seq.NEXTVAL, 10, '23-FEB-16');

INSERT INTO CustomerOrder (Customer\_Order\_ID, Customer\_ID, Order\_Date)

VALUES (CustomerOrder\_Seq.NEXTVAL, 11, '04-APR-16');

INSERT INTO CustomerOrder (Customer\_Order\_ID, Customer\_ID, Order\_Date)

VALUES (CustomerOrder\_Seq.NEXTVAL, 12, '15-MAR-16');

INSERT INTO CustomerOrder (Customer\_Order\_ID, Customer\_ID, Order\_Date)

VALUES (CustomerOrder\_Seq.NEXTVAL, 13, '16-MAR-16');

INSERT INTO CustomerOrder (Customer\_Order\_ID, Customer\_ID, Order\_Date)

VALUES (CustomerOrder\_Seq.NEXTVAL, 14, '15-JAN-16');

INSERT INTO CustomerOrder (Customer\_Order\_ID, Customer\_ID, Order\_Date)

VALUES (CustomerOrder\_Seq.NEXTVAL, 15, '10-JAN-16');

INSERT INTO CustomerOrder (Customer\_Order\_ID, Customer\_ID, Order\_Date)

VALUES (CustomerOrder\_Seq.NEXTVAL, 16, '01-MAR-16');

INSERT INTO CustomerOrder (Customer\_Order\_ID, Customer\_ID, Order\_Date)

VALUES (CustomerOrder\_Seq.NEXTVAL, 17, '03-JAN-16');

INSERT INTO CustomerOrder (Customer\_Order\_ID, Customer\_ID, Order\_Date)

VALUES (CustomerOrder\_Seq.NEXTVAL, 18, '20-MAR-16');

INSERT INTO CustomerOrder (Customer\_Order\_ID, Customer\_ID, Order\_Date)

VALUES (CustomerOrder\_Seq.NEXTVAL, 19, '22-DEC-15');

INSERT INTO CustomerOrder (Customer\_Order\_ID, Customer\_ID, Order\_Date)

VALUES (CustomerOrder\_Seq.NEXTVAL, 20, '23-FEB-16');

INSERT INTO ItemType (Item\_Type\_ID, Type\_Name)

VALUES (ItemType\_Seq.NEXTVAL, 'CPU');

INSERT INTO ItemType (Item\_Type\_ID, Type\_Name)

VALUES (ItemType\_Seq.NEXTVAL, 'Battery');

INSERT INTO ItemType (Item\_Type\_ID, Type\_Name)

VALUES (ItemType\_Seq.NEXTVAL, 'Desktop');

INSERT INTO ItemType (Item\_Type\_ID, Type\_Name)

VALUES (ItemType\_Seq.NEXTVAL, 'Laptop');

INSERT INTO ItemType (Item\_Type\_ID, Type\_Name)

VALUES (ItemType\_Seq.NEXTVAL, 'Monitor');

INSERT INTO ItemType (Item\_Type\_ID, Type\_Name)

VALUES (ItemType\_Seq.NEXTVAL, 'Printer');

INSERT INTO ItemType (Item\_Type\_ID, Type\_Name)

VALUES (ItemType\_Seq.NEXTVAL, 'RAM');

INSERT INTO ItemType (Item\_Type\_ID, Type\_Name)

VALUES (ItemType\_Seq.NEXTVAL, 'Floppy Drive');

INSERT INTO ItemType (Item\_Type\_ID, Type\_Name)

VALUES (ItemType\_Seq.NEXTVAL, 'DVD Drive');

INSERT INTO ItemType (Item\_Type\_ID, Type\_Name)

VALUES (ItemType\_Seq.NEXTVAL, 'Video Card');

INSERT INTO ItemType (Item\_Type\_ID, Type\_Name)

VALUES (ItemType\_Seq.NEXTVAL, 'GPU');

INSERT INTO ItemType (Item\_Type\_ID, Type\_Name)

VALUES (ItemType\_Seq.NEXTVAL, 'Keyboard');

INSERT INTO ItemType (Item\_Type\_ID, Type\_Name)

VALUES (ItemType\_Seq.NEXTVAL, 'Mouse');

INSERT INTO ItemType (Item\_Type\_ID, Type\_Name)

VALUES (ItemType\_Seq.NEXTVAL, 'BluRay Drive');

INSERT INTO ItemType (Item\_Type\_ID, Type\_Name)

VALUES (ItemType\_Seq.NEXTVAL, 'Motherboard');

INSERT INTO ItemType (Item\_Type\_ID, Type\_Name)

VALUES (ItemType\_Seq.NEXTVAL, 'Ink');

INSERT INTO ItemType (Item\_Type\_ID, Type\_Name)

VALUES (ItemType\_Seq.NEXTVAL, 'Toner');

INSERT INTO ItemType (Item\_Type\_ID, Type\_Name)

VALUES (ItemType\_Seq.NEXTVAL, 'Tape Drive');

INSERT INTO ItemType (Item\_Type\_ID, Type\_Name)

VALUES (ItemType\_Seq.NEXTVAL, 'Sound Card');

INSERT INTO ItemType (Item\_Type\_ID, Type\_Name)

VALUES (ItemType\_Seq.NEXTVAL, 'Network Card');

INSERT INTO Item (Item\_ID, Item\_ListPrice, Item\_Cost, Item\_Name, Item\_Type\_ID, Spec\_Documents, Total\_Qty, Reorder\_Qty, Date\_Added)

VALUES (Item\_Seq.NEXTVAL, 123.23, 50.12, '4.2GHZ Intel I5 6th Gen',1, EMPTY\_BLOB(), 100, 1, '01-FEB-16');

INSERT INTO Item (Item\_ID, Item\_ListPrice, Item\_Cost, Item\_Name, Item\_Type\_ID, Spec\_Documents, Total\_Qty, Reorder\_Qty, Date\_Added)

VALUES (Item\_Seq.NEXTVAL, 1000.01, 500.08, 'Dell XPS 13 Laptop', 4, EMPTY\_BLOB(), 100, 1, '01-JAN-16');

INSERT INTO Item (Item\_ID, Item\_ListPrice, Item\_Cost, Item\_Name, Item\_Type\_ID, Spec\_Documents, Total\_Qty, Reorder\_Qty, Date\_Added)

VALUES (Item\_Seq.NEXTVAL, 56.34, 12.12, 'A42-U53 Battery', 2, EMPTY\_BLOB(), 100, 10, '01-FEB-15');

INSERT INTO Item (Item\_ID, Item\_ListPrice, Item\_Cost, Item\_Name, Item\_Type\_ID, Spec\_Documents, Total\_Qty, Reorder\_Qty, Date\_Added)

VALUES (Item\_Seq.NEXTVAL, 450.25, 80.00, 'HP Pavilion Desktop', 3, EMPTY\_BLOB(), 50, 2, '01-APR-15');

INSERT INTO Item (Item\_ID, Item\_ListPrice, Item\_Cost, Item\_Name, Item\_Type\_ID, Spec\_Documents, Total\_Qty, Reorder\_Qty, Date\_Added)

VALUES (Item\_Seq.NEXTVAL, 100.25, 68.00, 'HP 22 Inch Monitor', 5, EMPTY\_BLOB(), 50, 6, '01-JUN-15');

INSERT INTO Item (Item\_ID, Item\_ListPrice, Item\_Cost, Item\_Name, Item\_Type\_ID, Spec\_Documents, Total\_Qty, Reorder\_Qty, Date\_Added)

VALUES (Item\_Seq.NEXTVAL, 75.25, 25.00, 'LexMark Printer 5500', 6, EMPTY\_BLOB(), 25, 2, '01-JAN-15');

INSERT INTO Item (Item\_ID, Item\_ListPrice, Item\_Cost, Item\_Name, Item\_Type\_ID, Spec\_Documents, Total\_Qty, Reorder\_Qty, Date\_Added)

VALUES (Item\_Seq.NEXTVAL, 40.00, 10.00, '4GB DDR3 144-Pin RAM', 7, EMPTY\_BLOB(), 25, 5, '01-JUL-15');

INSERT INTO Item (Item\_ID, Item\_ListPrice, Item\_Cost, Item\_Name, Item\_Type\_ID, Spec\_Documents, Total\_Qty, Reorder\_Qty, Date\_Added)

VALUES (Item\_Seq.NEXTVAL, 15.25, 3.00, 'Dell 3.5 Inch Floppy', 8, EMPTY\_BLOB(), 10, 3, '01-SEP-15');

INSERT INTO Item (Item\_ID, Item\_ListPrice, Item\_Cost, Item\_Name, Item\_Type\_ID, Spec\_Documents, Total\_Qty, Reorder\_Qty, Date\_Added)

VALUES (Item\_Seq.NEXTVAL, 55.25, 28.00, 'Dell 200x DVD Burner', 9, EMPTY\_BLOB(), 10, 2, '01-MAR-15');

INSERT INTO Item (Item\_ID, Item\_ListPrice, Item\_Cost, Item\_Name, Item\_Type\_ID, Spec\_Documents, Total\_Qty, Reorder\_Qty, Date\_Added)

VALUES (Item\_Seq.NEXTVAL, 300.00, 58.00, 'AMD EVGA 1GB DDR3 Video Card', 10, EMPTY\_BLOB(), 10, 1, '01-JUN-15');

INSERT INTO Item (Item\_ID, Item\_ListPrice, Item\_Cost, Item\_Name, Item\_Type\_ID, Spec\_Documents, Total\_Qty, Reorder\_Qty, Date\_Added)

VALUES (Item\_Seq.NEXTVAL, 125.23, 50.12, 'Intel GPU',11, EMPTY\_BLOB(), 100, 1, '01-FEB-16');

INSERT INTO Item (Item\_ID, Item\_ListPrice, Item\_Cost, Item\_Name, Item\_Type\_ID, Spec\_Documents, Total\_Qty, Reorder\_Qty, Date\_Added)

VALUES (Item\_Seq.NEXTVAL, 1001.01, 500.08, 'IBM Keyboard', 12, EMPTY\_BLOB(), 100, 1, '01-JAN-16');

INSERT INTO Item (Item\_ID, Item\_ListPrice, Item\_Cost, Item\_Name, Item\_Type\_ID, Spec\_Documents, Total\_Qty, Reorder\_Qty, Date\_Added)

VALUES (Item\_Seq.NEXTVAL, 55.34, 12.12, 'Dell Mouse', 13, EMPTY\_BLOB(), 100, 10, '01-FEB-15');

INSERT INTO Item (Item\_ID, Item\_ListPrice, Item\_Cost, Item\_Name, Item\_Type\_ID, Spec\_Documents, Total\_Qty, Reorder\_Qty, Date\_Added)

VALUES (Item\_Seq.NEXTVAL, 451.25, 80.00, 'Sony 50x speed BlueRay Drive', 14, EMPTY\_BLOB(), 50, 2, '01-APR-15');

INSERT INTO Item (Item\_ID, Item\_ListPrice, Item\_Cost, Item\_Name, Item\_Type\_ID, Spec\_Documents, Total\_Qty, Reorder\_Qty, Date\_Added)

VALUES (Item\_Seq.NEXTVAL, 103.25, 68.00, 'ASUS Motherboard 3.0', 15, EMPTY\_BLOB(), 50, 6, '01-JUN-15');

INSERT INTO Item (Item\_ID, Item\_ListPrice, Item\_Cost, Item\_Name, Item\_Type\_ID, Spec\_Documents, Total\_Qty, Reorder\_Qty, Date\_Added)

VALUES (Item\_Seq.NEXTVAL, 72.25, 25.00, 'LexMark 5500 Black Ink', 16, EMPTY\_BLOB(), 25, 2, '01-JAN-15');

INSERT INTO Item (Item\_ID, Item\_ListPrice, Item\_Cost, Item\_Name, Item\_Type\_ID, Spec\_Documents, Total\_Qty, Reorder\_Qty, Date\_Added)

VALUES (Item\_Seq.NEXTVAL, 41.00, 10.00, 'LexMark LaserJet Toner', 17, EMPTY\_BLOB(), 25, 5, '01-JUL-15');

INSERT INTO Item (Item\_ID, Item\_ListPrice, Item\_Cost, Item\_Name, Item\_Type\_ID, Spec\_Documents, Total\_Qty, Reorder\_Qty, Date\_Added)

VALUES (Item\_Seq.NEXTVAL, 16.25, 3.00, 'Dell Tape Drive', 18, EMPTY\_BLOB(), 10, 3, '01-SEP-15');

INSERT INTO Item (Item\_ID, Item\_ListPrice, Item\_Cost, Item\_Name, Item\_Type\_ID, Spec\_Documents, Total\_Qty, Reorder\_Qty, Date\_Added)

VALUES (Item\_Seq.NEXTVAL, 59.25, 28.00, 'Sony SoundCard', 19, EMPTY\_BLOB(), 10, 2, '01-MAR-15');

INSERT INTO Item (Item\_ID, Item\_ListPrice, Item\_Cost, Item\_Name, Item\_Type\_ID, Spec\_Documents, Total\_Qty, Reorder\_Qty, Date\_Added)

VALUES (Item\_Seq.NEXTVAL, 310.00, 58.00, 'IBM Network Card', 20, EMPTY\_BLOB(), 10, 1, '01-JUN-15');

INSERT INTO OrderDetails (Order\_Details\_ID, Order\_Details\_Qty, Item\_ID, Customer\_Order\_ID, Discount)

VALUES (OrderDetails\_Seq.NEXTVAL, 1, 1, 1, .05);

INSERT INTO OrderDetails (Order\_Details\_ID, Order\_Details\_Qty, Item\_ID, Customer\_Order\_ID, Discount)

VALUES (OrderDetails\_Seq.NEXTVAL, 2, 2, 1, .05);

INSERT INTO OrderDetails (Order\_Details\_ID, Order\_Details\_Qty, Item\_ID, Customer\_Order\_ID, Discount)

VALUES (OrderDetails\_Seq.NEXTVAL, 3, 3, 2, 0);

INSERT INTO OrderDetails (Order\_Details\_ID, Order\_Details\_Qty, Item\_ID, Customer\_Order\_ID, Discount)

VALUES (OrderDetails\_Seq.NEXTVAL, 4, 4, 3, .10);

INSERT INTO OrderDetails (Order\_Details\_ID, Order\_Details\_Qty, Item\_ID, Customer\_Order\_ID, Discount)

VALUES (OrderDetails\_Seq.NEXTVAL, 5, 5, 4, 0);

INSERT INTO OrderDetails (Order\_Details\_ID, Order\_Details\_Qty, Item\_ID, Customer\_Order\_ID, Discount)

VALUES (OrderDetails\_Seq.NEXTVAL, 6, 6, 5, 0);

INSERT INTO OrderDetails (Order\_Details\_ID, Order\_Details\_Qty, Item\_ID, Customer\_Order\_ID, Discount)

VALUES (OrderDetails\_Seq.NEXTVAL, 7, 7, 6, 0);

INSERT INTO OrderDetails (Order\_Details\_ID, Order\_Details\_Qty, Item\_ID, Customer\_Order\_ID, Discount)

VALUES (OrderDetails\_Seq.NEXTVAL, 8, 8, 7, 0);

INSERT INTO OrderDetails (Order\_Details\_ID, Order\_Details\_Qty, Item\_ID, Customer\_Order\_ID, Discount)

VALUES (OrderDetails\_Seq.NEXTVAL, 1, 9, 8, 0);

INSERT INTO OrderDetails (Order\_Details\_ID, Order\_Details\_Qty, Item\_ID, Customer\_Order\_ID, Discount)

VALUES (OrderDetails\_Seq.NEXTVAL, 1, 10, 9, 0);

INSERT INTO OrderDetails (Order\_Details\_ID, Order\_Details\_Qty, Item\_ID, Customer\_Order\_ID, Discount)

VALUES (OrderDetails\_Seq.NEXTVAL, 1, 11, 11, .05);

INSERT INTO OrderDetails (Order\_Details\_ID, Order\_Details\_Qty, Item\_ID, Customer\_Order\_ID, Discount)

VALUES (OrderDetails\_Seq.NEXTVAL, 2, 12, 12, .05);

INSERT INTO OrderDetails (Order\_Details\_ID, Order\_Details\_Qty, Item\_ID, Customer\_Order\_ID, Discount)

VALUES (OrderDetails\_Seq.NEXTVAL, 3, 13, 13, 0);

INSERT INTO OrderDetails (Order\_Details\_ID, Order\_Details\_Qty, Item\_ID, Customer\_Order\_ID, Discount)

VALUES (OrderDetails\_Seq.NEXTVAL, 4, 14, 14, .10);

INSERT INTO OrderDetails (Order\_Details\_ID, Order\_Details\_Qty, Item\_ID, Customer\_Order\_ID, Discount)

VALUES (OrderDetails\_Seq.NEXTVAL, 5, 15, 15, 0);

INSERT INTO OrderDetails (Order\_Details\_ID, Order\_Details\_Qty, Item\_ID, Customer\_Order\_ID, Discount)

VALUES (OrderDetails\_Seq.NEXTVAL, 6, 16, 16, 0);

INSERT INTO OrderDetails (Order\_Details\_ID, Order\_Details\_Qty, Item\_ID, Customer\_Order\_ID, Discount)

VALUES (OrderDetails\_Seq.NEXTVAL, 7, 17, 17, 0);

INSERT INTO OrderDetails (Order\_Details\_ID, Order\_Details\_Qty, Item\_ID, Customer\_Order\_ID, Discount)

VALUES (OrderDetails\_Seq.NEXTVAL, 8, 18, 18, 0);

INSERT INTO OrderDetails (Order\_Details\_ID, Order\_Details\_Qty, Item\_ID, Customer\_Order\_ID, Discount)

VALUES (OrderDetails\_Seq.NEXTVAL, 1, 19, 19, 0);

INSERT INTO OrderDetails (Order\_Details\_ID, Order\_Details\_Qty, Item\_ID, Customer\_Order\_ID, Discount)

VALUES (OrderDetails\_Seq.NEXTVAL, 1, 10, 20, 0);

commit;

/\* Verify each tables content \*/

SELECT /\*fixed\*/ \* FROM Customer;

SELECT /\*fixed\*/ \* FROM CustomerOrder;

SELECT /\*fixed\*/ \* FROM OrderDetails;

SELECT /\*fixed\*/ \* FROM Item;

SELECT /\*fixed\*/ \* FROM ItemType;

### Object Output/Log

Below is the output log for the DDL script processing.

SQL> /\* Insert data into each table \*/

SQL> INSERT INTO Customer (Customer\_ID, Cust\_FName, Cust\_LName, Cust\_Address, Cust\_ZipCode, Cust\_PhoneNum, Cust\_Email)

VALUES (Customer\_Seq.NEXTVAL, 'Bob', 'Roberts', '123 ABC Street','31088', '123-456-0000', 'BobRoberts@gmail.com');

1 row inserted.

SQL> INSERT INTO Customer (Customer\_ID, Cust\_FName, Cust\_LName, Cust\_Address, Cust\_ZipCode, Cust\_PhoneNum, Cust\_Email)

VALUES (Customer\_Seq.NEXTVAL, 'Mary', 'Jane', '45 Alberta Lane ','31771', '476-897-1111', 'MaryJane@gmail.com');

1 row inserted.

SQL> INSERT INTO Customer (Customer\_ID, Cust\_FName, Cust\_LName, Cust\_Address, Cust\_ZipCode, Cust\_PhoneNum, Cust\_Email)

VALUES (Customer\_Seq.NEXTVAL, 'Brandon', 'Russell', '123 Finegand Place ','31771', '234-123-222', 'HighFive34@gmail.com');

1 row inserted.

SQL> INSERT INTO Customer (Customer\_ID, Cust\_FName, Cust\_LName, Cust\_Address, Cust\_ZipCode, Cust\_PhoneNum, Cust\_Email)

VALUES (Customer\_Seq.NEXTVAL, 'James', 'Doe', '46 Alberta Lane ','31771', '476-897-1112', 'JamesDoe@gmail.com');

1 row inserted.

SQL> INSERT INTO Customer (Customer\_ID, Cust\_FName, Cust\_LName, Cust\_Address, Cust\_ZipCode, Cust\_PhoneNum, Cust\_Email)

VALUES (Customer\_Seq.NEXTVAL, 'Jim', 'Bob', '47 Alberta Lane ','31771', '476-897-1113', 'JimBob@gmail.com');

1 row inserted.

SQL> INSERT INTO Customer (Customer\_ID, Cust\_FName, Cust\_LName, Cust\_Address, Cust\_ZipCode, Cust\_PhoneNum, Cust\_Email)

VALUES (Customer\_Seq.NEXTVAL, 'Patricia', 'Russell', '48 Alberta Lane ','31771', '476-897-1114', 'PRussell@gmail.com');

1 row inserted.

SQL> INSERT INTO Customer (Customer\_ID, Cust\_FName, Cust\_LName, Cust\_Address, Cust\_ZipCode, Cust\_PhoneNum, Cust\_Email)

VALUES (Customer\_Seq.NEXTVAL, 'Jane', 'Johnson', '49 Alberta Lane ','31771', '476-897-1115', 'JJ123@gmail.com');

1 row inserted.

SQL> INSERT INTO Customer (Customer\_ID, Cust\_FName, Cust\_LName, Cust\_Address, Cust\_ZipCode, Cust\_PhoneNum, Cust\_Email)

VALUES (Customer\_Seq.NEXTVAL, 'Hyon', 'Yi', '50 Alberta Lane ','31771', '476-897-1116', 'HyonYi@gmail.com');

1 row inserted.

SQL> INSERT INTO Customer (Customer\_ID, Cust\_FName, Cust\_LName, Cust\_Address, Cust\_ZipCode, Cust\_PhoneNum, Cust\_Email)

VALUES (Customer\_Seq.NEXTVAL, 'Keith', 'Jordan', '51 Alberta Lane ','31771', '476-897-1117', 'KJ88@gmail.com');

1 row inserted.

SQL> INSERT INTO Customer (Customer\_ID, Cust\_FName, Cust\_LName, Cust\_Address, Cust\_ZipCode, Cust\_PhoneNum, Cust\_Email)

VALUES (Customer\_Seq.NEXTVAL, 'Pete', 'Smith', '52 Alberta Lane ','31771', '476-897-1118', 'PeteS33@gmail.com');

1 row inserted.

SQL> INSERT INTO Customer (Customer\_ID, Cust\_FName, Cust\_LName, Cust\_Address, Cust\_ZipCode, Cust\_PhoneNum, Cust\_Email)

VALUES (Customer\_Seq.NEXTVAL, 'Bobby', 'Robert', '124 ABC Street','31088', '223-456-0000', 'BobRobert@gmail.com');

1 row inserted.

SQL> INSERT INTO Customer (Customer\_ID, Cust\_FName, Cust\_LName, Cust\_Address, Cust\_ZipCode, Cust\_PhoneNum, Cust\_Email)

VALUES (Customer\_Seq.NEXTVAL, 'MaryJane', 'Joy', '46 Alberta Lane ','31771', '576-897-1111', 'MaryJane1@gmail.com');

1 row inserted.

SQL> INSERT INTO Customer (Customer\_ID, Cust\_FName, Cust\_LName, Cust\_Address, Cust\_ZipCode, Cust\_PhoneNum, Cust\_Email)

VALUES (Customer\_Seq.NEXTVAL, 'Brando', 'Russell', '124 Finegand Place ','31771', '334-123-222', 'H2ighFive34@gmail.com');

1 row inserted.

SQL> INSERT INTO Customer (Customer\_ID, Cust\_FName, Cust\_LName, Cust\_Address, Cust\_ZipCode, Cust\_PhoneNum, Cust\_Email)

VALUES (Customer\_Seq.NEXTVAL, 'Jay', 'Doe', '411 Alberta Lane ','31771', '471-897-1112', 'JDoe@gmail.com');

1 row inserted.

SQL> INSERT INTO Customer (Customer\_ID, Cust\_FName, Cust\_LName, Cust\_Address, Cust\_ZipCode, Cust\_PhoneNum, Cust\_Email)

VALUES (Customer\_Seq.NEXTVAL, 'Jimmie', 'Horton', '470 Alberta Lane ','31771', '476-897-1119', 'JimHorton@gmail.com');

1 row inserted.

SQL> INSERT INTO Customer (Customer\_ID, Cust\_FName, Cust\_LName, Cust\_Address, Cust\_ZipCode, Cust\_PhoneNum, Cust\_Email)

VALUES (Customer\_Seq.NEXTVAL, 'Patrick', 'Russell', '480 Alberta Lane ','31771', '476-897-1124', 'PRussell1@gmail.com');

1 row inserted.

SQL> INSERT INTO Customer (Customer\_ID, Cust\_FName, Cust\_LName, Cust\_Address, Cust\_ZipCode, Cust\_PhoneNum, Cust\_Email)

VALUES (Customer\_Seq.NEXTVAL, 'John', 'Johnson', '490 Alberta Lane ','31771', '476-897-1315', 'JJ124@gmail.com');

1 row inserted.

SQL> INSERT INTO Customer (Customer\_ID, Cust\_FName, Cust\_LName, Cust\_Address, Cust\_ZipCode, Cust\_PhoneNum, Cust\_Email)

VALUES (Customer\_Seq.NEXTVAL, 'Hyon', 'Yie', '500 Alberta Lane ','31771', '476-897-1516', 'HyonYi1@gmail.com');

1 row inserted.

SQL> INSERT INTO Customer (Customer\_ID, Cust\_FName, Cust\_LName, Cust\_Address, Cust\_ZipCode, Cust\_PhoneNum, Cust\_Email)

VALUES (Customer\_Seq.NEXTVAL, 'Mike', 'Jordan', '510 Alberta Lane ','31771', '476-997-1117', 'MJ88@gmail.com');

1 row inserted.

SQL> INSERT INTO Customer (Customer\_ID, Cust\_FName, Cust\_LName, Cust\_Address, Cust\_ZipCode, Cust\_PhoneNum, Cust\_Email)

VALUES (Customer\_Seq.NEXTVAL, 'Peter', 'Smith', '520 Alberta Lane ','31771', '476-899-1118', 'PeterS33@gmail.com');

1 row inserted.

SQL> INSERT INTO CustomerOrder (Customer\_Order\_ID, Customer\_ID, Order\_Date)

VALUES (CustomerOrder\_Seq.NEXTVAL, 1, '12-FEB-15');

1 row inserted.

SQL> INSERT INTO CustomerOrder (Customer\_Order\_ID, Customer\_ID, Order\_Date)

VALUES (CustomerOrder\_Seq.NEXTVAL, 2, '15-FEB-16');

1 row inserted.

SQL> INSERT INTO CustomerOrder (Customer\_Order\_ID, Customer\_ID, Order\_Date)

VALUES (CustomerOrder\_Seq.NEXTVAL, 3, '16-FEB-16');

1 row inserted.

SQL> INSERT INTO CustomerOrder (Customer\_Order\_ID, Customer\_ID, Order\_Date)

VALUES (CustomerOrder\_Seq.NEXTVAL, 4, '15-JAN-16');

1 row inserted.

SQL> INSERT INTO CustomerOrder (Customer\_Order\_ID, Customer\_ID, Order\_Date)

VALUES (CustomerOrder\_Seq.NEXTVAL, 5, '10-JAN-16');

1 row inserted.

SQL> INSERT INTO CustomerOrder (Customer\_Order\_ID, Customer\_ID, Order\_Date)

VALUES (CustomerOrder\_Seq.NEXTVAL, 6, '01-FEB-16');

1 row inserted.

SQL> INSERT INTO CustomerOrder (Customer\_Order\_ID, Customer\_ID, Order\_Date)

VALUES (CustomerOrder\_Seq.NEXTVAL, 7, '03-DEC-15');

1 row inserted.

SQL> INSERT INTO CustomerOrder (Customer\_Order\_ID, Customer\_ID, Order\_Date)

VALUES (CustomerOrder\_Seq.NEXTVAL, 8, '20-NOV-15');

1 row inserted.

SQL> INSERT INTO CustomerOrder (Customer\_Order\_ID, Customer\_ID, Order\_Date)

VALUES (CustomerOrder\_Seq.NEXTVAL, 9, '22-OCT-15');

1 row inserted.

SQL> INSERT INTO CustomerOrder (Customer\_Order\_ID, Customer\_ID, Order\_Date)

VALUES (CustomerOrder\_Seq.NEXTVAL, 10, '23-FEB-16');

1 row inserted.

SQL> INSERT INTO CustomerOrder (Customer\_Order\_ID, Customer\_ID, Order\_Date)

VALUES (CustomerOrder\_Seq.NEXTVAL, 11, '04-APR-16');

1 row inserted.

SQL> INSERT INTO CustomerOrder (Customer\_Order\_ID, Customer\_ID, Order\_Date)

VALUES (CustomerOrder\_Seq.NEXTVAL, 12, '15-MAR-16');

1 row inserted.

SQL> INSERT INTO CustomerOrder (Customer\_Order\_ID, Customer\_ID, Order\_Date)

VALUES (CustomerOrder\_Seq.NEXTVAL, 13, '16-MAR-16');

1 row inserted.

SQL> INSERT INTO CustomerOrder (Customer\_Order\_ID, Customer\_ID, Order\_Date)

VALUES (CustomerOrder\_Seq.NEXTVAL, 14, '15-JAN-16');

1 row inserted.

SQL> INSERT INTO CustomerOrder (Customer\_Order\_ID, Customer\_ID, Order\_Date)

VALUES (CustomerOrder\_Seq.NEXTVAL, 15, '10-JAN-16');

1 row inserted.

SQL> INSERT INTO CustomerOrder (Customer\_Order\_ID, Customer\_ID, Order\_Date)

VALUES (CustomerOrder\_Seq.NEXTVAL, 16, '01-MAR-16');

1 row inserted.

SQL> INSERT INTO CustomerOrder (Customer\_Order\_ID, Customer\_ID, Order\_Date)

VALUES (CustomerOrder\_Seq.NEXTVAL, 17, '03-JAN-16');

1 row inserted.

SQL> INSERT INTO CustomerOrder (Customer\_Order\_ID, Customer\_ID, Order\_Date)

VALUES (CustomerOrder\_Seq.NEXTVAL, 18, '20-MAR-16');

1 row inserted.

SQL> INSERT INTO CustomerOrder (Customer\_Order\_ID, Customer\_ID, Order\_Date)

VALUES (CustomerOrder\_Seq.NEXTVAL, 19, '22-DEC-15');

1 row inserted.

SQL> INSERT INTO CustomerOrder (Customer\_Order\_ID, Customer\_ID, Order\_Date)

VALUES (CustomerOrder\_Seq.NEXTVAL, 20, '23-FEB-16');

1 row inserted.

SQL> INSERT INTO ItemType (Item\_Type\_ID, Type\_Name)

VALUES (ItemType\_Seq.NEXTVAL, 'CPU');

1 row inserted.

SQL> INSERT INTO ItemType (Item\_Type\_ID, Type\_Name)

VALUES (ItemType\_Seq.NEXTVAL, 'Battery');

1 row inserted.

SQL> INSERT INTO ItemType (Item\_Type\_ID, Type\_Name)

VALUES (ItemType\_Seq.NEXTVAL, 'Desktop');

1 row inserted.

SQL> INSERT INTO ItemType (Item\_Type\_ID, Type\_Name)

VALUES (ItemType\_Seq.NEXTVAL, 'Laptop');

1 row inserted.

SQL> INSERT INTO ItemType (Item\_Type\_ID, Type\_Name)

VALUES (ItemType\_Seq.NEXTVAL, 'Monitor');

1 row inserted.

SQL> INSERT INTO ItemType (Item\_Type\_ID, Type\_Name)

VALUES (ItemType\_Seq.NEXTVAL, 'Printer');

1 row inserted.

SQL> INSERT INTO ItemType (Item\_Type\_ID, Type\_Name)

VALUES (ItemType\_Seq.NEXTVAL, 'RAM');

1 row inserted.

SQL> INSERT INTO ItemType (Item\_Type\_ID, Type\_Name)

VALUES (ItemType\_Seq.NEXTVAL, 'Floppy Drive');

1 row inserted.

SQL> INSERT INTO ItemType (Item\_Type\_ID, Type\_Name)

VALUES (ItemType\_Seq.NEXTVAL, 'DVD Drive');

1 row inserted.

SQL> INSERT INTO ItemType (Item\_Type\_ID, Type\_Name)

VALUES (ItemType\_Seq.NEXTVAL, 'Video Card');

1 row inserted.

SQL> INSERT INTO ItemType (Item\_Type\_ID, Type\_Name)

VALUES (ItemType\_Seq.NEXTVAL, 'GPU');

1 row inserted.

SQL> INSERT INTO ItemType (Item\_Type\_ID, Type\_Name)

VALUES (ItemType\_Seq.NEXTVAL, 'Keyboard');

1 row inserted.

SQL> INSERT INTO ItemType (Item\_Type\_ID, Type\_Name)

VALUES (ItemType\_Seq.NEXTVAL, 'Mouse');

1 row inserted.

SQL> INSERT INTO ItemType (Item\_Type\_ID, Type\_Name)

VALUES (ItemType\_Seq.NEXTVAL, 'BluRay Drive');

1 row inserted.

SQL> INSERT INTO ItemType (Item\_Type\_ID, Type\_Name)

VALUES (ItemType\_Seq.NEXTVAL, 'Motherboard');

1 row inserted.

SQL> INSERT INTO ItemType (Item\_Type\_ID, Type\_Name)

VALUES (ItemType\_Seq.NEXTVAL, 'Ink');

1 row inserted.

SQL> INSERT INTO ItemType (Item\_Type\_ID, Type\_Name)

VALUES (ItemType\_Seq.NEXTVAL, 'Toner');

1 row inserted.

SQL> INSERT INTO ItemType (Item\_Type\_ID, Type\_Name)

VALUES (ItemType\_Seq.NEXTVAL, 'Tape Drive');

1 row inserted.

SQL> INSERT INTO ItemType (Item\_Type\_ID, Type\_Name)

VALUES (ItemType\_Seq.NEXTVAL, 'Sound Card');

1 row inserted.

SQL> INSERT INTO ItemType (Item\_Type\_ID, Type\_Name)

VALUES (ItemType\_Seq.NEXTVAL, 'Network Card');

1 row inserted.

SQL> INSERT INTO Item (Item\_ID, Item\_ListPrice, Item\_Cost, Item\_Name, Item\_Type\_ID, Spec\_Documents, Total\_Qty, Reorder\_Qty, Date\_Added)

VALUES (Item\_Seq.NEXTVAL, 123.23, 50.12, '4.2GHZ Intel I5 6th Gen',1, EMPTY\_BLOB(), 100, 1, '01-FEB-16');

1 row inserted.

SQL> INSERT INTO Item (Item\_ID, Item\_ListPrice, Item\_Cost, Item\_Name, Item\_Type\_ID, Spec\_Documents, Total\_Qty, Reorder\_Qty, Date\_Added)

VALUES (Item\_Seq.NEXTVAL, 1000.01, 500.08, 'Dell XPS 13 Laptop', 4, EMPTY\_BLOB(), 100, 1, '01-JAN-16');

1 row inserted.

SQL> INSERT INTO Item (Item\_ID, Item\_ListPrice, Item\_Cost, Item\_Name, Item\_Type\_ID, Spec\_Documents, Total\_Qty, Reorder\_Qty, Date\_Added)

VALUES (Item\_Seq.NEXTVAL, 56.34, 12.12, 'A42-U53 Battery', 2, EMPTY\_BLOB(), 100, 10, '01-FEB-15');

1 row inserted.

SQL> INSERT INTO Item (Item\_ID, Item\_ListPrice, Item\_Cost, Item\_Name, Item\_Type\_ID, Spec\_Documents, Total\_Qty, Reorder\_Qty, Date\_Added)

VALUES (Item\_Seq.NEXTVAL, 450.25, 80.00, 'HP Pavilion Desktop', 3, EMPTY\_BLOB(), 50, 2, '01-APR-15');

1 row inserted.

SQL> INSERT INTO Item (Item\_ID, Item\_ListPrice, Item\_Cost, Item\_Name, Item\_Type\_ID, Spec\_Documents, Total\_Qty, Reorder\_Qty, Date\_Added)

VALUES (Item\_Seq.NEXTVAL, 100.25, 68.00, 'HP 22 Inch Monitor', 5, EMPTY\_BLOB(), 50, 6, '01-JUN-15');

1 row inserted.

SQL> INSERT INTO Item (Item\_ID, Item\_ListPrice, Item\_Cost, Item\_Name, Item\_Type\_ID, Spec\_Documents, Total\_Qty, Reorder\_Qty, Date\_Added)

VALUES (Item\_Seq.NEXTVAL, 75.25, 25.00, 'LexMark Printer 5500', 6, EMPTY\_BLOB(), 25, 2, '01-JAN-15');

1 row inserted.

SQL> INSERT INTO Item (Item\_ID, Item\_ListPrice, Item\_Cost, Item\_Name, Item\_Type\_ID, Spec\_Documents, Total\_Qty, Reorder\_Qty, Date\_Added)

VALUES (Item\_Seq.NEXTVAL, 40.00, 10.00, '4GB DDR3 144-Pin RAM', 7, EMPTY\_BLOB(), 25, 5, '01-JUL-15');

1 row inserted.

SQL> INSERT INTO Item (Item\_ID, Item\_ListPrice, Item\_Cost, Item\_Name, Item\_Type\_ID, Spec\_Documents, Total\_Qty, Reorder\_Qty, Date\_Added)

VALUES (Item\_Seq.NEXTVAL, 15.25, 3.00, 'Dell 3.5 Inch Floppy', 8, EMPTY\_BLOB(), 10, 3, '01-SEP-15');

1 row inserted.

SQL> INSERT INTO Item (Item\_ID, Item\_ListPrice, Item\_Cost, Item\_Name, Item\_Type\_ID, Spec\_Documents, Total\_Qty, Reorder\_Qty, Date\_Added)

VALUES (Item\_Seq.NEXTVAL, 55.25, 28.00, 'Dell 200x DVD Burner', 9, EMPTY\_BLOB(), 10, 2, '01-MAR-15');

1 row inserted.

SQL> INSERT INTO Item (Item\_ID, Item\_ListPrice, Item\_Cost, Item\_Name, Item\_Type\_ID, Spec\_Documents, Total\_Qty, Reorder\_Qty, Date\_Added)

VALUES (Item\_Seq.NEXTVAL, 300.00, 58.00, 'AMD EVGA 1GB DDR3 Video Card', 10, EMPTY\_BLOB(), 10, 1, '01-JUN-15');

1 row inserted.

SQL> INSERT INTO Item (Item\_ID, Item\_ListPrice, Item\_Cost, Item\_Name, Item\_Type\_ID, Spec\_Documents, Total\_Qty, Reorder\_Qty, Date\_Added)

VALUES (Item\_Seq.NEXTVAL, 125.23, 50.12, 'Intel GPU',11, EMPTY\_BLOB(), 100, 1, '01-FEB-16');

1 row inserted.

SQL> INSERT INTO Item (Item\_ID, Item\_ListPrice, Item\_Cost, Item\_Name, Item\_Type\_ID, Spec\_Documents, Total\_Qty, Reorder\_Qty, Date\_Added)

VALUES (Item\_Seq.NEXTVAL, 1001.01, 500.08, 'IBM Keyboard', 12, EMPTY\_BLOB(), 100, 1, '01-JAN-16');

1 row inserted.

SQL> INSERT INTO Item (Item\_ID, Item\_ListPrice, Item\_Cost, Item\_Name, Item\_Type\_ID, Spec\_Documents, Total\_Qty, Reorder\_Qty, Date\_Added)

VALUES (Item\_Seq.NEXTVAL, 55.34, 12.12, 'Dell Mouse', 13, EMPTY\_BLOB(), 100, 10, '01-FEB-15');

1 row inserted.

SQL> INSERT INTO Item (Item\_ID, Item\_ListPrice, Item\_Cost, Item\_Name, Item\_Type\_ID, Spec\_Documents, Total\_Qty, Reorder\_Qty, Date\_Added)

VALUES (Item\_Seq.NEXTVAL, 451.25, 80.00, 'Sony 50x speed BlueRay Drive', 14, EMPTY\_BLOB(), 50, 2, '01-APR-15');

1 row inserted.

SQL> INSERT INTO Item (Item\_ID, Item\_ListPrice, Item\_Cost, Item\_Name, Item\_Type\_ID, Spec\_Documents, Total\_Qty, Reorder\_Qty, Date\_Added)

VALUES (Item\_Seq.NEXTVAL, 103.25, 68.00, 'ASUS Motherboard 3.0', 15, EMPTY\_BLOB(), 50, 6, '01-JUN-15');

1 row inserted.

SQL> INSERT INTO Item (Item\_ID, Item\_ListPrice, Item\_Cost, Item\_Name, Item\_Type\_ID, Spec\_Documents, Total\_Qty, Reorder\_Qty, Date\_Added)

VALUES (Item\_Seq.NEXTVAL, 72.25, 25.00, 'LexMark 5500 Black Ink', 16, EMPTY\_BLOB(), 25, 2, '01-JAN-15');

1 row inserted.

SQL> INSERT INTO Item (Item\_ID, Item\_ListPrice, Item\_Cost, Item\_Name, Item\_Type\_ID, Spec\_Documents, Total\_Qty, Reorder\_Qty, Date\_Added)

VALUES (Item\_Seq.NEXTVAL, 41.00, 10.00, 'LexMark LaserJet Toner', 17, EMPTY\_BLOB(), 25, 5, '01-JUL-15');

1 row inserted.

SQL> INSERT INTO Item (Item\_ID, Item\_ListPrice, Item\_Cost, Item\_Name, Item\_Type\_ID, Spec\_Documents, Total\_Qty, Reorder\_Qty, Date\_Added)

VALUES (Item\_Seq.NEXTVAL, 16.25, 3.00, 'Dell Tape Drive', 18, EMPTY\_BLOB(), 10, 3, '01-SEP-15');

1 row inserted.

SQL> INSERT INTO Item (Item\_ID, Item\_ListPrice, Item\_Cost, Item\_Name, Item\_Type\_ID, Spec\_Documents, Total\_Qty, Reorder\_Qty, Date\_Added)

VALUES (Item\_Seq.NEXTVAL, 59.25, 28.00, 'Sony SoundCard', 19, EMPTY\_BLOB(), 10, 2, '01-MAR-15');

1 row inserted.

SQL> INSERT INTO Item (Item\_ID, Item\_ListPrice, Item\_Cost, Item\_Name, Item\_Type\_ID, Spec\_Documents, Total\_Qty, Reorder\_Qty, Date\_Added)

VALUES (Item\_Seq.NEXTVAL, 310.00, 58.00, 'IBM Network Card', 20, EMPTY\_BLOB(), 10, 1, '01-JUN-15');

1 row inserted.

SQL> INSERT INTO OrderDetails (Order\_Details\_ID, Order\_Details\_Qty, Item\_ID, Customer\_Order\_ID, Discount)

VALUES (OrderDetails\_Seq.NEXTVAL, 1, 1, 1, .05);

1 row inserted.

SQL> INSERT INTO OrderDetails (Order\_Details\_ID, Order\_Details\_Qty, Item\_ID, Customer\_Order\_ID, Discount)

VALUES (OrderDetails\_Seq.NEXTVAL, 2, 2, 1, .05);

1 row inserted.

SQL> INSERT INTO OrderDetails (Order\_Details\_ID, Order\_Details\_Qty, Item\_ID, Customer\_Order\_ID, Discount)

VALUES (OrderDetails\_Seq.NEXTVAL, 3, 3, 2, 0);

1 row inserted.

SQL> INSERT INTO OrderDetails (Order\_Details\_ID, Order\_Details\_Qty, Item\_ID, Customer\_Order\_ID, Discount)

VALUES (OrderDetails\_Seq.NEXTVAL, 4, 4, 3, .10);

1 row inserted.

SQL> INSERT INTO OrderDetails (Order\_Details\_ID, Order\_Details\_Qty, Item\_ID, Customer\_Order\_ID, Discount)

VALUES (OrderDetails\_Seq.NEXTVAL, 5, 5, 4, 0);

1 row inserted.

SQL> INSERT INTO OrderDetails (Order\_Details\_ID, Order\_Details\_Qty, Item\_ID, Customer\_Order\_ID, Discount)

VALUES (OrderDetails\_Seq.NEXTVAL, 6, 6, 5, 0);

1 row inserted.

SQL> INSERT INTO OrderDetails (Order\_Details\_ID, Order\_Details\_Qty, Item\_ID, Customer\_Order\_ID, Discount)

VALUES (OrderDetails\_Seq.NEXTVAL, 7, 7, 6, 0);

1 row inserted.

SQL> INSERT INTO OrderDetails (Order\_Details\_ID, Order\_Details\_Qty, Item\_ID, Customer\_Order\_ID, Discount)

VALUES (OrderDetails\_Seq.NEXTVAL, 8, 8, 7, 0);

1 row inserted.

SQL> INSERT INTO OrderDetails (Order\_Details\_ID, Order\_Details\_Qty, Item\_ID, Customer\_Order\_ID, Discount)

VALUES (OrderDetails\_Seq.NEXTVAL, 1, 9, 8, 0);

1 row inserted.

SQL> INSERT INTO OrderDetails (Order\_Details\_ID, Order\_Details\_Qty, Item\_ID, Customer\_Order\_ID, Discount)

VALUES (OrderDetails\_Seq.NEXTVAL, 1, 10, 9, 0);

1 row inserted.

SQL> INSERT INTO OrderDetails (Order\_Details\_ID, Order\_Details\_Qty, Item\_ID, Customer\_Order\_ID, Discount)

VALUES (OrderDetails\_Seq.NEXTVAL, 1, 11, 11, .05);

1 row inserted.

SQL> INSERT INTO OrderDetails (Order\_Details\_ID, Order\_Details\_Qty, Item\_ID, Customer\_Order\_ID, Discount)

VALUES (OrderDetails\_Seq.NEXTVAL, 2, 12, 12, .05);

1 row inserted.

SQL> INSERT INTO OrderDetails (Order\_Details\_ID, Order\_Details\_Qty, Item\_ID, Customer\_Order\_ID, Discount)

VALUES (OrderDetails\_Seq.NEXTVAL, 3, 13, 13, 0);

1 row inserted.

SQL> INSERT INTO OrderDetails (Order\_Details\_ID, Order\_Details\_Qty, Item\_ID, Customer\_Order\_ID, Discount)

VALUES (OrderDetails\_Seq.NEXTVAL, 4, 14, 14, .10);

1 row inserted.

SQL> INSERT INTO OrderDetails (Order\_Details\_ID, Order\_Details\_Qty, Item\_ID, Customer\_Order\_ID, Discount)

VALUES (OrderDetails\_Seq.NEXTVAL, 5, 15, 15, 0);

1 row inserted.

SQL> INSERT INTO OrderDetails (Order\_Details\_ID, Order\_Details\_Qty, Item\_ID, Customer\_Order\_ID, Discount)

VALUES (OrderDetails\_Seq.NEXTVAL, 6, 16, 16, 0);

1 row inserted.

SQL> INSERT INTO OrderDetails (Order\_Details\_ID, Order\_Details\_Qty, Item\_ID, Customer\_Order\_ID, Discount)

VALUES (OrderDetails\_Seq.NEXTVAL, 7, 17, 17, 0);

1 row inserted.

SQL> INSERT INTO OrderDetails (Order\_Details\_ID, Order\_Details\_Qty, Item\_ID, Customer\_Order\_ID, Discount)

VALUES (OrderDetails\_Seq.NEXTVAL, 8, 18, 18, 0);

1 row inserted.

SQL> INSERT INTO OrderDetails (Order\_Details\_ID, Order\_Details\_Qty, Item\_ID, Customer\_Order\_ID, Discount)

VALUES (OrderDetails\_Seq.NEXTVAL, 1, 19, 19, 0);

1 row inserted.

SQL> INSERT INTO OrderDetails (Order\_Details\_ID, Order\_Details\_Qty, Item\_ID, Customer\_Order\_ID, Discount)

VALUES (OrderDetails\_Seq.NEXTVAL, 1, 10, 20, 0);

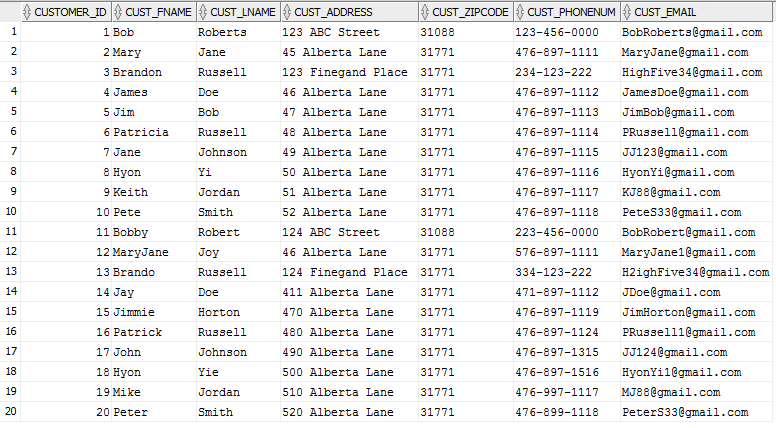
1 row inserted.

SQL> commit;

Commit complete.

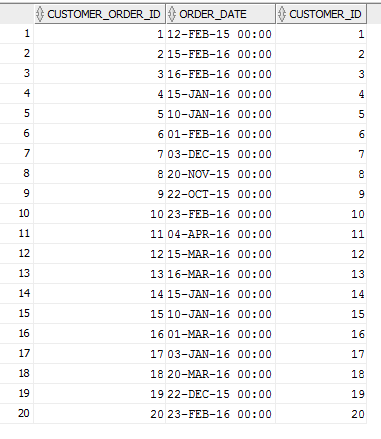
SQL> /\* Verify each tables content \*/

SQL> SELECT /\*fixed\*/ \* FROM Customer;



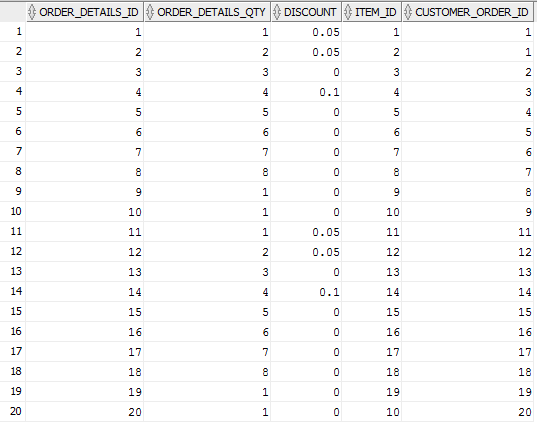
20 rows selected

SQL> SELECT /\*fixed\*/ \* FROM CustomerOrder;



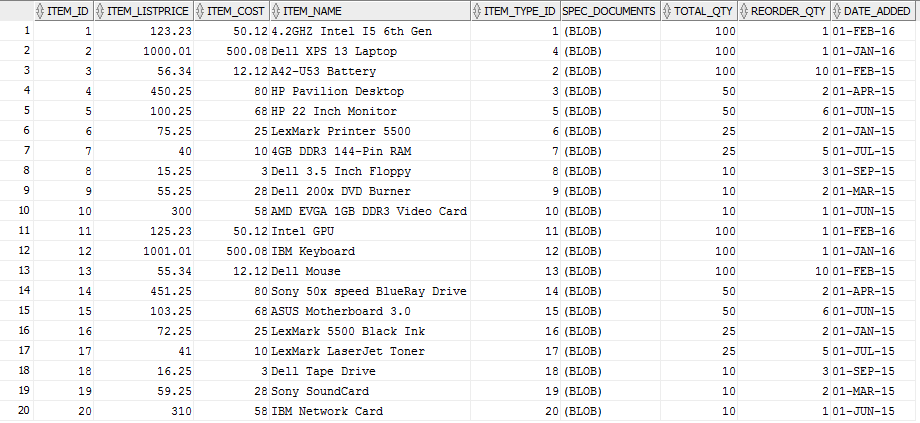
20 rows selected

SQL> SELECT /\*fixed\*/ \* FROM OrderDetails;



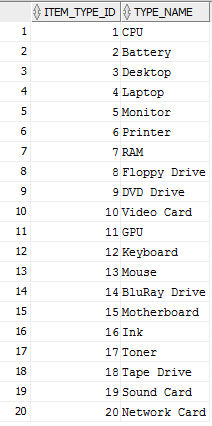
20 rows selected

SQL> SELECT /\*fixed\*/ \* FROM Item;



20 rows selected

SQL> SELECT /\*fixed\*/ \* FROM ItemType;



20 rows selected

# SQL Queries (Code & Output)

# Simple Queries

1. The below query generates basic details about all orders in the database. This includes information about the customer, the order quantity, order date, and item details.

SELECT

c.CUSTOMER\_ID,

c.CUST\_FNAME,

c.CUST\_LNAME,

o.ORDER\_DATE,

d.ORDER\_DETAILS\_QTY,

d.DISCOUNT,

i.ITEM\_LISTPRICE,

i.ITEM\_COST,

i.ITEM\_NAME,

t.TYPE\_NAME,

ROUND((i.ITEM\_LISTPRICE - (d.DISCOUNT \* i.ITEM\_LISTPRICE)),2) AS Sale\_price

FROM

CUSTOMER c,

CUSTOMERORDER o,

ORDERDETAILS d,

ITEM i,

ITEMTYPE t

WHERE

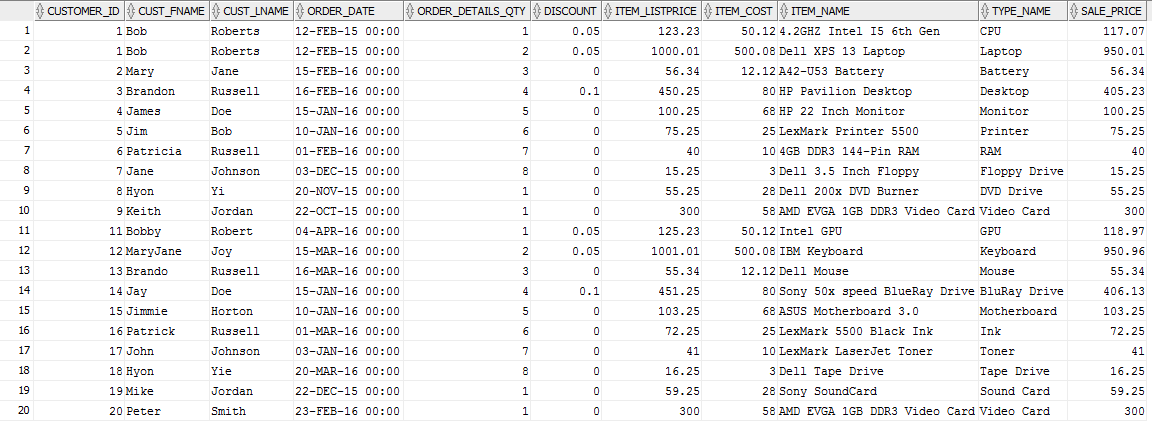
c.CUSTOMER\_ID = o.CUSTOMER\_ID and

o.CUSTOMER\_ORDER\_ID = d.CUSTOMER\_ORDER\_ID and

d.ITEM\_ID = i.ITEM\_ID and

i.ITEM\_TYPE\_ID = t.ITEM\_TYPE\_ID;

**Output Table**



1. Below is an example of how certain reports could be generated to determine if a customer should be targeted for sale of a new product. In this example customers for whom 1 year has passed since their last CPU purchase are selected.

SELECT

c.CUSTOMER\_ID,

c.CUST\_FNAME,

c.CUST\_LNAME,

o.ORDER\_DATE,

d.ORDER\_DETAILS\_QTY,

i.ITEM\_NAME,

t.TYPE\_NAME

FROM

CUSTOMER c,

CUSTOMERORDER o,

ORDERDETAILS d,

ITEM i,

ITEMTYPE t

WHERE

c.CUSTOMER\_ID = o.CUSTOMER\_ID and

o.CUSTOMER\_ORDER\_ID = d.CUSTOMER\_ORDER\_ID and

d.ITEM\_ID = i.ITEM\_ID and

i.ITEM\_TYPE\_ID = t.ITEM\_TYPE\_ID and

t.ITEM\_TYPE\_ID = 1 and

o.ORDER\_DATE < ADD\_MONTHS(SYSDATE, -12);

**Output Table**



# Advanced Queries

1. The below query reports the number of items of each product type that have sold

since the product was added to the database compared with the current date, the date

the product was added, and finally the number of days’ difference between the two

SELECT

i.item\_name,

a.count AS sold\_since\_added,

i.DATE\_ADDED,

SYSDATE, (

ROUND((SYSDATE - i.date\_added),0)) AS Difference\_in\_days

FROM

ITEM i,

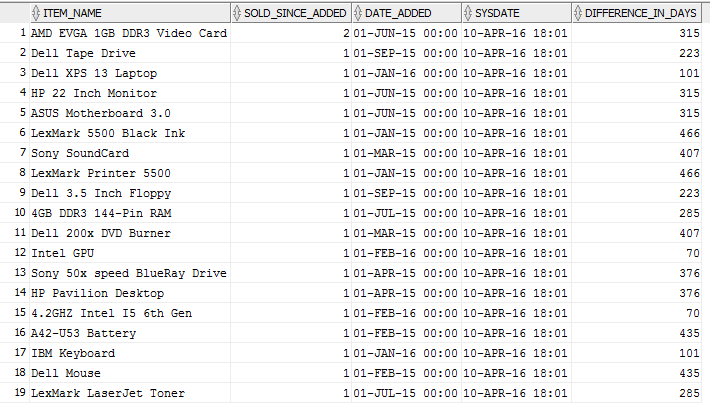
(SELECT COUNT(d.item\_id) AS count, d.item\_id AS itemid

FROM orderdetails d

GROUP BY d.ITEM\_ID) a

WHERE a.itemid = i.item\_id;

**Output Table**



1. The below query reports the most recent sale of each product compared with the difference in days of that date and the current system date

SELECT

d.Item\_ID,

o.Order\_Date AS LAST\_DATE\_ORDERED,

i.Item\_Name,

(ROUND((SYSDATE - o.Order\_Date),0)) AS DAYS\_SINCE\_LAST\_ORDER

FROM

OrderDetails d

INNER JOIN

CustomerOrder o ON o.customer\_order\_id=d.customer\_order\_id

INNER JOIN

Item i ON i.Item\_ID=d.Item\_ID

WHERE

o.Order\_Date IN (SELECT MAX(x.Order\_Date)

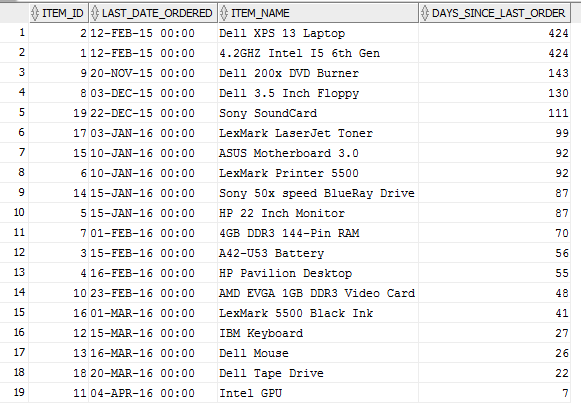
FROM CustomerOrder x

INNER JOIN OrderDetails y ON y.customer\_order\_id=x.customer\_order\_id

WHERE y.Item\_ID = d.Item\_ID )

ORDER BY DAYS\_SINCE\_LAST\_ORDER DESC;

**Output Table**



# Misc Queries

1. This query uses concatenation to connect two strings together to become one. This produces a result of a single column.

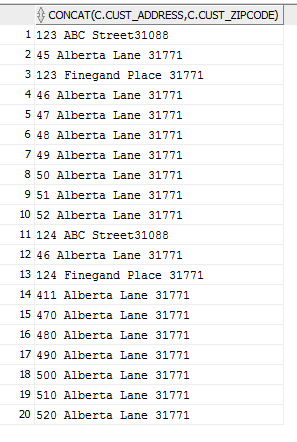
Without comma seperation

\*/

SELECT CONCAT (c.cust\_address, c.cust\_zipcode)

FROM customer c;

**Output Table**

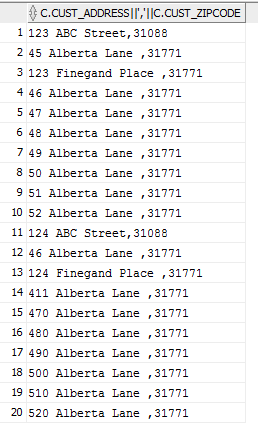


-- With comma separation

SELECT c.CUST\_ADDRESS||','|| c.CUST\_ZIPCODE

FROM CUSTOMER c;

**Output Table**

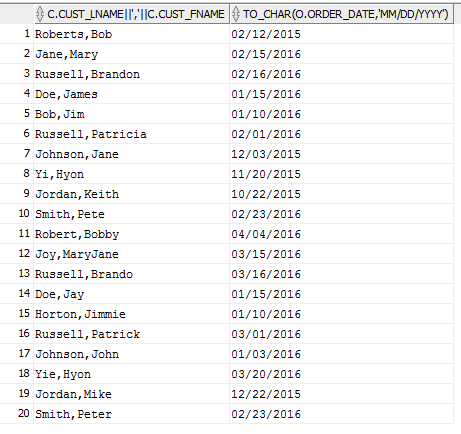


1. The query below concatenates the name and reformats the date from "DD-MON-YY" to "MM/DD/YYYY"

SELECT c.CUST\_LNAME||','||c.CUST\_FNAME,TO\_CHAR(o.ORDER\_DATE, 'MM/DD/YYYY')

FROM CUSTOMER c INNER JOIN CUSTOMERORDER o ON c.CUSTOMER\_ID = o.CUSTOMER\_ID;

**Output Table**



1. The query below compiles the average cost of all items sold and the average discount that was received from the purchases within a date range.

SELECT ROUND(AVG(t.ITEM\_COST), 2) AS "Average Item Cost", AVG(d.DISCOUNT \* 100) AS "Discount in Percent"

FROM CUSTOMERORDER o INNER JOIN ORDERDETAILS d

ON o.CUSTOMER\_ORDER\_ID = d.CUSTOMER\_ORDER\_ID

INNER JOIN ITEM t ON d.ITEM\_ID = t.ITEM\_ID

WHERE o.ORDER\_DATE BETWEEN '15-JAN-10' AND '17-JAN-17';

**Output Table**



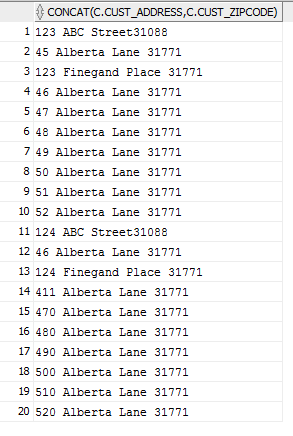
1. This query uses concatenation to connect two strings together to become one. This produces a result of a single column.

Without comma seperation

Select CONCAT (c.cust\_address, c.cust\_zipcode)

from customer c;

**Output Table**

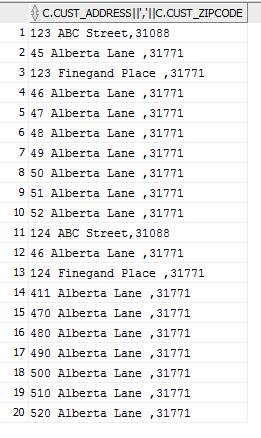


With comma separation

Select C.CUST\_ADDRESS||','|| C.CUST\_ZIPCODE

FROM CUSTOMER C;

**Output Table**



# Database Maintenance

Database maintenance tasks to boost performance in searches and add some functionality

## Index Code

Adding indexes on commonly searched fields

CREATE INDEX in\_orderdate on CustomerOrder(Order\_Date);

CREATE INDEX in\_cost on Item(Item\_Cost);

## Stored Procedure Code

Stored procedure to deduct inventory after customer purchases. This will be invoked by the front-end application, and is an alternative to a trigger. Performance would improve over a trigger due to it not being run after every update on table ITEM

CREATE OR REPLACE PROCEDURE remove\_inventory

(item\_qty IN NUMBER

, item\_id IN NUMBER

)

AS

BEGIN

UPDATE ITEM

SET Total\_Qty = Total\_Qty - remove\_inventory.item\_qty

WHERE Item\_ID = remove\_inventory.item\_id;

END;

/

/\*Production cost reduced for the following items\*/

UPDATE item

SET Item\_ListPrice=101.23,

Item\_Cost=28.12

WHERE Item\_ID=1;

/\*Update item quantity based on recent inventory audits\*/

UPDATE item

SET Total\_Qty=500

WHERE Item\_ID=2;

/\*Remove items from inventory and associated orders\*/

DELETE FROM OrderDetails WHERE Order\_Details\_ID=3;

DELETE FROM Item WHERE Item\_ID=3;

/\*Insert new orders from backlog\*/

INSERT INTO OrderDetails (Order\_Details\_ID, Order\_Details\_Qty, Item\_ID, Customer\_Order\_ID, Discount)

VALUES (2001, 1, 10, 8, 0);

INSERT INTO OrderDetails (Order\_Details\_ID, Order\_Details\_Qty, Item\_ID, Customer\_Order\_ID, Discount)

VALUES (2002, 1, 10, 7, 0);

## Alter Command Code

Rename an index to improve code readability

ALTER INDEX fk\_Customer RENAME TO fk\_Customer\_Order\_Index;

/\*Verify changes\*/

SELECT /\*fixed\*/ \* FROM Item;

SELECT /\*fixed\*/ \* FROM User\_Indexes WHERE Index\_Name LIKE 'FK%' OR Index\_Name LIKE 'IN%';

# Explain Tool

/\*

Group 3

RLS DBST652 - Using the Explain plan tool to look at query execution plans.

With Oracle we can see that one needs to store the execution plan of a particular query

and then use database function to recall that plan. By setting the 'STATEMENT\_ID' we are

able to set and recall specific ones.

Depending on the option you input to display them will depend on what is displayed, but you can

see that some will give you the Cost of each particular action in the query (in terms of CPU power)

and then the time it takes to process.

You also have the option of using the Oracle Timing command, or '.get\_time' procedure found within the database

utility package. All of these will be shown as examples below, with their results displayed.

\*/

-- Using the Explain tool to store the Explain plan data into the default table PLAN\_TABLE

EXPLAIN PLAN

SET STATEMENT\_ID = 'GroupPlan\_1' FOR

SELECT

d.Item\_ID,

o.Order\_Date AS LAST\_DATE\_ORDERED,

i.Item\_Name,

(ROUND((SYSDATE - o.Order\_Date),0)) AS DAYS\_SINCE\_LAST\_ORDER

FROM

OrderDetails d

INNER JOIN

CustomerOrder o ON o.customer\_order\_id=d.customer\_order\_id

INNER JOIN

Item i ON i.Item\_ID=d.Item\_ID

WHERE

o.Order\_Date IN (SELECT MAX(x.Order\_Date)

FROM CustomerOrder x

INNER JOIN OrderDetails y ON y.customer\_order\_id=x.customer\_order\_id

WHERE y.Item\_ID = d.Item\_ID )

ORDER BY DAYS\_SINCE\_LAST\_ORDER DESC;

EXPLAIN PLAN

SET STATEMENT\_ID = 'GroupPlan\_2' FOR

SELECT Customer\_ID, Cust\_FName

FROM Customer

UNION

SELECT Customer\_ID, TO\_CHAR(Order\_date)

FROM CustomerOrder;

EXPLAIN PLAN

SET STATEMENT\_ID = 'GroupPlan\_3' FOR

SELECT

i.item\_name,

a.count AS sold\_since\_added,

i.DATE\_ADDED,

SYSDATE, (

ROUND((SYSDATE - i.date\_added),0)) AS Difference\_in\_days

FROM

ITEM i,

(SELECT COUNT(d.item\_id) AS count, d.item\_id AS itemid

FROM orderdetails d

GROUP BY d.ITEM\_ID) a

WHERE a.itemid = i.item\_id;

-- Display results from Explain PLAN\_TABLE

-- With 'Basic' output parameter

SELECT PLAN\_TABLE\_OUTPUT FROM TABLE(DBMS\_XPLAN.DISPLAY('PLAN\_TABLE','GroupPlan\_1','BASIC'));

-- Results

/\*

Plan hash value: 2492395418

----------------------------------------------------------------

| Id | Operation | Name |

----------------------------------------------------------------

| 0 | SELECT STATEMENT | |

| 1 | SORT ORDER BY | |

| 2 | NESTED LOOPS | |

| 3 | NESTED LOOPS | |

| 4 | HASH JOIN | |

| 5 | HASH JOIN | |

| 6 | VIEW | VW\_SQ\_1 |

| 7 | HASH GROUP BY | |

| 8 | MERGE JOIN | |

| 9 | TABLE ACCESS BY INDEX ROWID| CUSTOMERORDER |

| 10 | INDEX FULL SCAN | PK\_CUSTOMERORDER |

| 11 | SORT JOIN | |

| 12 | VIEW | index$\_join$\_007 |

| 13 | HASH JOIN | |

| 14 | INDEX FAST FULL SCAN | FK\_CUSTOMERORDER |

| 15 | INDEX FAST FULL SCAN | FK\_ORDERITEM |

| 16 | VIEW | index$\_join$\_001 |

| 17 | HASH JOIN | |

| 18 | INDEX FAST FULL SCAN | FK\_CUSTOMERORDER |

| 19 | INDEX FAST FULL SCAN | FK\_ORDERITEM |

| 20 | VIEW | index$\_join$\_002 |

| 21 | HASH JOIN | |

| 22 | INDEX FAST FULL SCAN | IN\_ORDERDATE |

| 23 | INDEX FAST FULL SCAN | PK\_CUSTOMERORDER |

| 24 | INDEX UNIQUE SCAN | PK\_ITEM |

| 25 | TABLE ACCESS BY INDEX ROWID | ITEM |

----------------------------------------------------------------

\*/

-- With 'Typical' output parameter

SELECT PLAN\_TABLE\_OUTPUT FROM TABLE(DBMS\_XPLAN.DISPLAY('PLAN\_TABLE','GroupPlan\_2','TYPICAL'));

--Results

/\*

Plan hash value: 1777963098

-----------------------------------------------------------------------------------

| Id | Operation | Name | Rows | Bytes | Cost (%CPU)| Time |

-----------------------------------------------------------------------------------

| 0 | SELECT STATEMENT | | 20 | 120 | 6 (50)| 00:00:01 |

| 1 | UNION-ALL | | | | | |

| 2 | TABLE ACCESS FULL| ITEM | 10 | 60 | 3 (0)| 00:00:01 |

| 3 | TABLE ACCESS FULL| ORDERDETAILS | 10 | 60 | 3 (0)| 00:00:01 |

-----------------------------------------------------------------------------------

\*/

-- With 'All' output parameter

SELECT PLAN\_TABLE\_OUTPUT FROM TABLE(DBMS\_XPLAN.DISPLAY('PLAN\_TABLE','GroupPlan\_3','ALL'));

-- Results

/\*

Plan hash value: 1872350007

---------------------------------------------------------------------------------------------

| Id | Operation | Name | Rows | Bytes | Cost (%CPU)| Time |

---------------------------------------------------------------------------------------------

| 0 | SELECT STATEMENT | | 10 | 590 | 4 (25)| 00:00:01 |

| 1 | MERGE JOIN | | 10 | 590 | 4 (25)| 00:00:01 |

| 2 | TABLE ACCESS BY INDEX ROWID| ITEM | 10 | 330 | 2 (0)| 00:00:01 |

| 3 | INDEX FULL SCAN | PK\_ITEM | 10 | | 1 (0)| 00:00:01 |

|\* 4 | SORT JOIN | | 10 | 260 | 2 (50)| 00:00:01 |

| 5 | VIEW | | 10 | 260 | 1 (0)| 00:00:01 |

| 6 | HASH GROUP BY | | 10 | 30 | 1 (0)| 00:00:01 |

| 7 | INDEX FULL SCAN | FK\_ORDERITEM | 10 | 30 | 1 (0)| 00:00:01 |

---------------------------------------------------------------------------------------------

Query Block Name / Object Alias (identified by operation id):

-------------------------------------------------------------

1 - SEL$1

2 - SEL$1 / I@SEL$1

3 - SEL$1 / I@SEL$1

5 - SEL$2 / A@SEL$1

6 - SEL$2

7 - SEL$2 / D@SEL$2

Predicate Information (identified by operation id):

---------------------------------------------------

4 - access("A"."ITEMID"="I"."ITEM\_ID")

filter("A"."ITEMID"="I"."ITEM\_ID")

Column Projection Information (identified by operation id):

-----------------------------------------------------------

1 - (#keys=0) "I"."DATE\_ADDED"[DATE,7], "I"."ITEM\_NAME"[VARCHAR2,50],

"A"."COUNT"[NUMBER,22]

2 - "I"."ITEM\_ID"[NUMBER,22], "I"."ITEM\_NAME"[VARCHAR2,50],

"I"."DATE\_ADDED"[DATE,7]

3 - "I".ROWID[ROWID,10], "I"."ITEM\_ID"[NUMBER,22]

4 - (#keys=1) "A"."ITEMID"[NUMBER,22], "A"."COUNT"[NUMBER,22]

5 - "A"."COUNT"[NUMBER,22], "A"."ITEMID"[NUMBER,22]

6 - (#keys=1) "D"."ITEM\_ID"[NUMBER,22], COUNT(\*)[22]

7 - "D"."ITEM\_ID"[NUMBER,22]

\*/

-- Using Oracle Timing command to get \*JUST\* time of query execution

SET serveroutput ON;

VARIABLE n NUMBER;

EXEC :n := dbms\_utility.get\_time;

SELECT

d.Item\_ID,

o.Order\_Date AS LAST\_DATE\_ORDERED,

i.Item\_Name,

(ROUND((SYSDATE - o.Order\_Date),0)) AS DAYS\_SINCE\_LAST\_ORDER

FROM

OrderDetails d

INNER JOIN

CustomerOrder o ON o.customer\_order\_id=d.customer\_order\_id

INNER JOIN

Item i ON i.Item\_ID=d.Item\_ID

WHERE

o.Order\_Date IN (SELECT MAX(x.Order\_Date)

FROM CustomerOrder x

INNER JOIN OrderDetails y ON y.customer\_order\_id=x.customer\_order\_id

WHERE y.Item\_ID = d.Item\_ID )

ORDER BY DAYS\_SINCE\_LAST\_ORDER DESC;

EXEC dbms\_output.put\_line(( (dbms\_utility.get\_time-:n)/100) || ' seconds....');

-- Result

/\*

anonymous block completed

ITEM\_ID LAST\_DATE\_ORDERED ITEM\_NAME DAYS\_SINCE\_LAST\_ORDER

---------- ----------------- -------------------------------------------------- ---------------------

10 22-OCT-15 AMD EVGA 1GB DDR3 Video Card 162

9 20-NOV-15 Dell 200x DVD Burner 133

8 03-DEC-15 Dell 3.5 Inch Floppy 120

6 10-JAN-16 LexMark Printer 5500 82

5 15-JAN-16 HP 22 Inch Monitor 77

7 01-FEB-16 4GB DDR3 144-Pin RAM 60

1 12-FEB-16 4.2GHZ Intel I5 6th Gen 49

2 12-FEB-16 Dell XPS 13 Laptop 49

3 15-FEB-16 A42-U53 Battery 46

4 16-FEB-16 HP Pavilion Desktop 45

10 rows selected

anonymous block completed

.02 seconds....

\*/

# References

Elmasri, R., & Shamkant, N. B. (2015). *Fundamentals of database systems.* Boston: Pearson.

Nikolov, p. (2011). *Aggregate queries in nosql cloud data stores.* Amsterdam: Department of Sciences, Vrije Universiteit, Amsterdam The Netherlands.

Rischert, A. (2010). *Oracle sql.* Upper Saddle River: Pearson Education.