Table of Contents

[ERD 2](#_Toc2620010)

[Calculations for Sizing Objects 2](#_Toc2620011)

[Data Definition Language (DDL) Scripts 6](#_Toc2620012)

[Drop Objects 6](#_Toc2620013)

[Create Users 12](#_Toc2620014)

[Drop Tablespaces 17](#_Toc2620015)

[Create Tablespaces and Alter Users 18](#_Toc2620016)

[Create Tables and Indexes 21](#_Toc2620017)

[Load Data 32](#_Toc2620018)

[SQL Control File Loaders 32](#_Toc2620019)

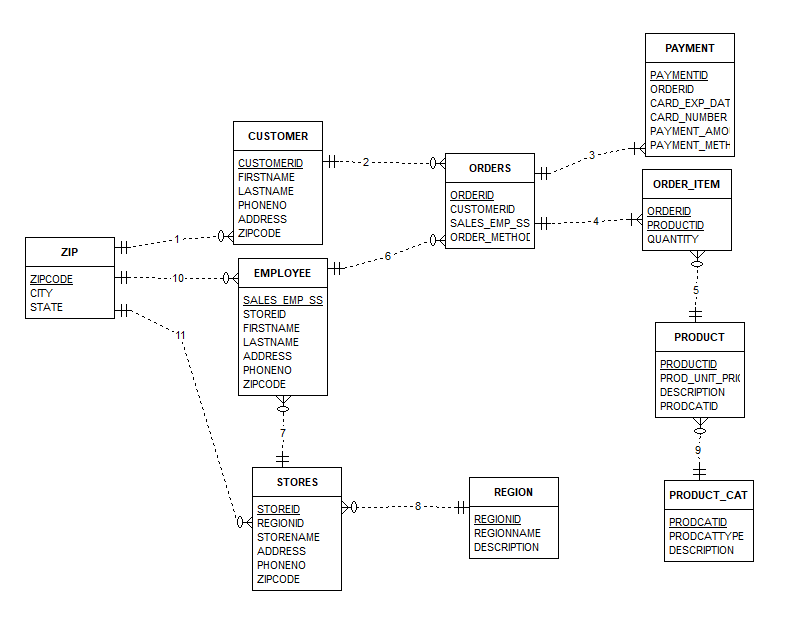
[Query Tables 44](#_Toc2620020)

[Analyze Tables 50](#_Toc2620021)

[Check Object Status in Catalog 56](#_Toc2620022)

[Query Row Numbers from Catalog 57](#_Toc2620023)

# **ERD**



# **Calculations for Sizing Objects**

UserData 1 = 2184K + 120K + 100K+120K+96K+192K+192K = 3,004K

UserData 2 = 2184K + 168K + 16K+40K = 2,408K

|  |  |  |
| --- | --- | --- |
| Zip 5/90  \_\_\_\_\_\_\_  2476K - Split in User\_Data1&2  Zipcode = 10 Varchar  City = 25 Varchar  State = 2 Char  data space = (8108 - 2\*R) - ((8108 - 2\*R) \* 5/100)  data space = (8108 - 2\*R) - (405.4 - .1\*R)  data space = 8108 - 2\*R - 405.4 + .1\*R  data space = 7702.6 - 1.9\*R  Total Bytes = 40  7702.6 - 1.9\*R/ 40  192.57 - 0.048R  1.048R = 192.57  R = 183.75  ceiling(50000/183.75) = ceiling(273)  273\*8K blocks  = 2184K | Customer Size 20/40  -----  120K - User\_Data1  CUSTOMERID = Integer = log(500)/Log(2)/8 = 1  Firstname = 15 varchar  Lastname = 15 varchar  phoneno = 15 char  address = 35 varchar  zipcode 10 char  data space = (8108 - 2\*R) - ((8108 - 2\*R) \* 20/100)  data space = (8108 - 2\*R) - (1621.6 - .4\*R)  data space = 8108 - 2\*R - 1621.6 + .4\*R  data space = 6486.4 - 1.6\*R  Total Bytes = 94  6486.4 - 1.6\*R/ 94  69 - 0.017R  1.017R = 69  R = 67.85  ceiling(1000/67.85) = ceiling(15)  15\*8K blocks  = 120 Kilobytes | Orders 5/60  -----  100K - User\_Data1  OrderID = Integer = log(500)/Log(2)/8 = 1  CustomerId = Integer = log(500)/Log(2)/8 = 1  Sales\_emp\_ssn = Integer log(199,606,147)/log(2)/8 = 4  Order\_Method = 10 char  data space = (8108 - 2\*R) - ((8108 - 2\*R) \* 5/100)  data space = (8108 - 2\*R) - (405.4 - .1\*R)  data space = 8108 - 2\*R - 405.4 + .1\*R  data space = 7702.6 - 1.9\*R  Total Bytes = 19  7702.6 - 1.9\*R/ 19  405.4 - 0.001R  1.001R = 405.4  R = 405  ceiling(1000/405) = ceiling(3)  3\*8K blocks  = 24K kilobytes |

|  |  |  |
| --- | --- | --- |
| Product 5/90  ----  150K - User\_Data2  ProductId = Integer = Log(7000)/Log(2)/8 = 2  Prod\_Unit\_Price = Log(80)/Log(2)/8 = 1  Description = 150 varchar  data space = (8108 - 2\*R) - ((8108 - 2\*R) \* 5/100)  data space = (8108 - 2\*R) - (405.4 - .1\*R)  data space = 8108 - 2\*R - 405.4 + .1\*R  data space = 7702.6 - 1.9\*R  Total Bytes = 156  7702.6 - 1.9\*R/ 156  49.38 - 0.01R  1.010R = 49.38  R = 48.89  ceiling(1000/49) = ceiling(21)  21\*8K blocks  = 168 Kilobytes | Order\_Item 5/90  ------  50K - User\_Data2  Orderid = Integer = log(500)/Log(2)/8 = 1  ProductID = Integer = log(500)/Log(2)/8 = 1  Quantity = Integer = log(2)/Log(2)/8 = 1  data space = (8108 - 2\*R) - ((8108 - 2\*R) \* 5/100)  data space = (8108 - 2\*R) - (405.4 - .1\*R)  data space = 8108 - 2\*R - 405.4 + .1\*R  data space = 7702.6 - 1.9\*R  Total Bytes = 6  7702.6 - 1.9\*R/ 6  1283.77 - 0.32R  1.31R = 1283.77  R = 979.98  ceiling(1000/980) = ceiling(2)  2\*8K blocks  = 16 Kilobytes | Payment 20/40  -----  100K - User\_Data 2  PaymentId = Integer = log(500)/Log(2)/8 = 1  OrderID = Integer = log(500)/Log(2)/8 = 1  Car\_Exp\_Date = date = 7  card\_number = Integer = log(1851851849647260)/log(2)/8 = 7  payment\_amount = Integer = Log(80)/Log(2)/8 = 1  payment\_method = 5 varchar  data space = (8108 - 2\*R) - ((8108 - 2\*R) \* 20/100)  data space = (8108 - 2\*R) - (1621.6 - .4\*R)  data space = 8108 - 2\*R - 1621.6 + .4\*R  data space = 6486.4 - 1.6\*R  Total Bytes = 25  6486.4 - 1.6\*R/ 25  259.46 - 0.064R  1.064R = 259.46  R = 243.9  ceiling(1000/243.9) = ceiling(5)  5\*8K blocks  = 40 Kilobytes |

|  |  |  |
| --- | --- | --- |
| Employee – UserData1  -------  20/40...like Customer table  Address = 35 Varchar  Firstname = 15 Varchar  Lastname = 15 Varchar  PhoneNo = 15 Varchar  Sales\_emp\_ssn = Integer = log(199,606,147)/log(2)/8 = 4  storeid - Integer = avg(500) = 1  zipcode = char 5  ---  data space = (8108 - 2\*R) - ((8108 - 2\*R) \* 20/100)  data space = (8108 - 2\*R) - (1621.6 - .4\*R)  data space = 8108 - 2\*R - 1621.6 + .4\*R  data space = 6486.4 - 1.6\*R  Total Bytes = 93  6486.4 - 1.6\*R/ 93  69.74 - 0.017R  1.017R = 69.74  R = 68.57  ceiling(1000/15) = ceiling(15)  15\*8K blocks  = 120 Kilobytes | Store – UserData1  -----  5/90  Address = 35  phoneno= 15  regionid = int = avg(500) = 1  storename =varchar 25  zipcode = char 5  data space = (8108 - 2\*R) - ((8108 - 2\*R) \* 5/100)  data space = (8108 - 2\*R) - (405.4 - .1\*R)  data space = 8108 - 2\*R - 405.4 + .1\*R  data space = 7702.6 - 1.9\*R  Total Bytes = 84  7702.6 - 1.9\*R/ 84  91.7 - 0.022R  1.022R = 91.7  R = 89.72  ceiling(1000/90) = ceiling(12)  12\*8K blocks  = 96K kilobytes | Region – UserData1  ------  5/90  description = varchar 150  regionid = int = avg(500) = 1  regionname = vharchar 25  data space = (8108 - 2\*R) - ((8108 - 2\*R) \* 5/100)  data space = (8108 - 2\*R) - (405.4 - .1\*R)  data space = 8108 - 2\*R - 405.4 + .1\*R  data space = 7702.6 - 1.9\*R  Total Bytes = 179  7702.6 - 1.9\*R/ 179  43.03 - 0.01R  1.010R = 43.03  R = 42.6  ceiling(1000/43) = ceiling(24)  12\*8K blocks  = 192 Kilobytes |

|  |
| --- |
| ProdCat – UserData1  ------  5/90  description = varchar 150  prodcatid = int = avg(500) = 1  prodcattype = varchar 25  data space = (8108 - 2\*R) - ((8108 - 2\*R) \* 5/100)  data space = (8108 - 2\*R) - (405.4 - .1\*R)  data space = 8108 - 2\*R - 405.4 + .1\*R  data space = 7702.6 - 1.9\*R  Total Bytes = 179  7702.6 - 1.9\*R/ 179  43.03 - 0.01R  1.010R = 43.03  R = 42.6  ceiling(1000/43) = ceiling(24)  12\*8K blocks  = 192 Kilobytes |

# **Data Definition Language (DDL) Scripts**

## **Drop Objects**

SQL> SET ECHO ON;

SQL>

SQL> SET SERVEROUTPUT ON;

SQL>

SQL> /\*---------------------------------------------------\*/

SQL> /\* DROP OBJECTS \*/

SQL> /\*---------------------------------------------------\*/

SQL>

SQL> SHOW USER;

USER is "SYSTEM"

SQL>

SQL> CONNECT DataDesignLeadUser/abc123;

Connected.

SQL>

SQL> SHOW USER;

USER is "DATADESIGNLEADUSER"

SQL>

SQL> /\*Drop Indexes\*/

SQL>

SQL> DROP INDEX fk\_zip;

Index FK\_ZIP dropped.

SQL>

SQL> DROP INDEX fk\_customerid;

Index FK\_CUSTOMERID dropped.

SQL>

SQL> DROP INDEX fk\_order\_id1;

Index FK\_ORDER\_ID1 dropped.

SQL>

SQL> DROP INDEX fk\_product\_id;

Index FK\_PRODUCT\_ID dropped.

SQL>

SQL> DROP INDEX fk\_order\_id2;

Index FK\_ORDER\_ID2 dropped.

SQL>

SQL> DROP INDEX fk\_emp\_storeid;

Index FK\_EMP\_STOREID dropped.

SQL>

SQL> DROP INDEX fk\_emp\_zip;

Index FK\_EMP\_ZIP dropped.

SQL>

SQL> DROP INDEX fk\_store\_region;

Index FK\_STORE\_REGION dropped.

SQL>

SQL> DROP INDEX fk\_store\_zip;

Index FK\_STORE\_ZIP dropped.

SQL>

SQL> DROP INDEX fk\_prodcatid;

Index FK\_PRODCATID dropped.

SQL>

SQL> DROP INDEX fk\_emp\_ssn;

Index FK\_EMP\_SSN dropped.

SQL>

SQL> /\*Drop Tables\*/

SQL>

SQL> DROP TABLE payment;

Table PAYMENT dropped.

SQL>

SQL> DROP TABLE order\_item;

Table ORDER\_ITEM dropped.

SQL>

SQL> DROP TABLE orders;

Table ORDERS dropped.

SQL>

SQL> DROP TABLE product;

Table PRODUCT dropped.

SQL>

SQL> DROP TABLE customer;

Table CUSTOMER dropped.

SQL>

SQL> DROP TABLE employee;

Table EMPLOYEE dropped.

SQL>

SQL> DROP TABLE stores;

Table STORES dropped.

SQL>

SQL> DROP TABLE zip;

Table ZIP dropped.

SQL>

SQL> DROP TABLE region;

Table REGION dropped.

SQL>

SQL> DROP TABLE product\_cat;

Table PRODUCT\_CAT dropped.

SQL>

SQL>

SQL> /\*Disconnect DataDesignLeadUser, Login as system to drop users and create other objects\*/

SQL>

SQL> DISCONNECT;

Disconnected from Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production

SQL>

SQL> SHOW USER;

USER is ""

SQL>

SQL> CONNECT system/sysdba;

Connected.

SQL>

SQL> SHOW USER;

USER is "SYSTEM"

SQL>

SQL> /\*Drop Users\*/

SQL>

SQL> DROP USER jkamando CASCADE;

User JKAMANDO dropped.

SQL>

SQL> DROP USER tokusanya CASCADE;

User TOKUSANYA dropped.

SQL>

SQL> DROP USER joluwadamilare CASCADE;

User JOLUWADAMILARE dropped.

SQL>

SQL> DROP USER brussell CASCADE;

User BRUSSELL dropped.

SQL>

SQL> DROP USER rseyoum CASCADE;

User RSEYOUM dropped.

SQL>

SQL> DROP USER dbauser CASCADE;

User DBAUSER dropped.

SQL>

SQL> DROP USER datadesignleaduser CASCADE;

User DATADESIGNLEADUSER dropped.

## **Create Users**

SQL>

SQL> /\*---------------------------------------------------\*/

SQL> /\* CREATE USERS \*/

SQL> /\*---------------------------------------------------\*/

SQL>

SQL> /\*Create DBA User and grant necessary permissions\*/

SQL> CREATE USER dbauser IDENTIFIED BY abc123

2 DEFAULT TABLESPACE users

3 TEMPORARY TABLESPACE temp;

User DBAUSER created.

SQL>

SQL> GRANT dba TO dbauser;

Grant succeeded.

SQL>

SQL> /\*Disconnect from System and connect to DBAUser\*/

SQL> SELECT 'alter system kill session '''

2 || sid

3 || ','

4 || serial#

5 || ''';'

6 FROM v$session

7 WHERE username = 'SYSTEM';

'ALTERSYSTEMKILLSESSION'''||SID||','||SERIAL#||''';'

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

alter system kill session '388,50053';

alter system kill session '394,35833';

SQL>

SQL> DISCONNECT;

Disconnected from Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production

SQL>

SQL> SHOW USER;

USER is ""

SQL>

SQL> CONNECT DBAUser/abc123;

Connected.

SQL>

SQL> SHOW USER;

USER is "DBAUSER"

SQL>

SQL>

SQL> CREATE USER jkamando IDENTIFIED BY abc123

2 DEFAULT TABLESPACE users

3 TEMPORARY TABLESPACE temp;

User JKAMANDO created.

SQL>

SQL> CREATE USER tokusanya IDENTIFIED BY abc123

2 DEFAULT TABLESPACE users

3 TEMPORARY TABLESPACE temp;

User TOKUSANYA created.

SQL>

SQL> CREATE USER joluwadamilare IDENTIFIED BY abc123

2 DEFAULT TABLESPACE users

3 TEMPORARY TABLESPACE temp;

User JOLUWADAMILARE created.

SQL>

SQL> CREATE USER brussell IDENTIFIED BY abc123

2 DEFAULT TABLESPACE users

3 TEMPORARY TABLESPACE temp;

User BRUSSELL created.

SQL>

SQL> CREATE USER rseyoum IDENTIFIED BY abc123

2 DEFAULT TABLESPACE users

3 TEMPORARY TABLESPACE temp;

User RSEYOUM created.

SQL>

SQL>

SQL> CREATE USER datadesignleaduser IDENTIFIED BY abc123

2 DEFAULT TABLESPACE users

3 TEMPORARY TABLESPACE temp;

User DATADESIGNLEADUSER created.

SQL>

SQL> /\*Grant necessary permissions\*/

SQL>

SQL> GRANT connect,resource TO jkamando;

Grant succeeded.

SQL>

SQL> GRANT connect,resource TO tokusanya;

Grant succeeded.

SQL>

SQL> GRANT connect,resource TO joluwadamilare;

Grant succeeded.

SQL>

SQL> GRANT connect,resource TO brussell;

Grant succeeded.

SQL>

SQL> GRANT connect,resource TO rseyoum;

Grant succeeded.

SQL>

SQL> GRANT connect,resource TO datadesignleaduser;

Grant succeeded.

SQL>

SQL> GRANT UNLIMITED TABLESPACE TO datadesignleaduser;

Grant succeeded.

## **Drop Tablespaces**

SQL>

SQL> /\*DROP TABLESPACES\*/

SQL>

SQL> DROP TABLESPACE "USER\_DATA1" INCLUDING CONTENTS;

TABLESPACE "USER\_DATA1" dropped.

SQL>

SQL> DROP TABLESPACE "USER\_DATA2" INCLUDING CONTENTS;

TABLESPACE "USER\_DATA2" dropped.

SQL>

SQL> DROP TABLESPACE "USER\_INDX1" INCLUDING CONTENTS;

TABLESPACE "USER\_INDX1" dropped.

SQL>

SQL> DROP TABLESPACE "USER\_TEMP" INCLUDING CONTENTS;

TABLESPACE "USER\_TEMP" dropped.

## **Create Tablespaces and Alter Users**

SQL>

SQL> /\*---------------------------------------------------\*/

SQL> /\* CREATE TABLESPACE AND ASSIGN TO USERS \*/

SQL> /\*---------------------------------------------------\*/

SQL>

SQL> CREATE TABLESPACE "USER\_DATA1" LOGGING DATAFILE '/u01/app/oracle/oradata/DBST670/user\_data101.dbf' SIZE 3004K REUSE AUTOEXTEND ON NEXT 512K MAXSIZE 65536K EXTENT MANAGEMENT LOCAL;

TABLESPACE "USER\_DATA1" created.

SQL> CREATE TABLESPACE "USER\_DATA2" LOGGING DATAFILE '/u01/app/oracle/oradata/DBST670/user\_data201.dbf' SIZE 2408K REUSE AUTOEXTEND ON NEXT 512K MAXSIZE 65536K EXTENT MANAGEMENT LOCAL;

TABLESPACE "USER\_DATA2" created.

SQL> CREATE TABLESPACE "USER\_INDX1" LOGGING DATAFILE '/u01/app/oracle/oradata/DBST670/user\_indx01.dbf' SIZE 4210K REUSE AUTOEXTEND ON NEXT 512K MAXSIZE 65536K EXTENT MANAGEMENT LOCAL;

TABLESPACE "USER\_INDX1" created.

SQL> CREATE TEMPORARY TABLESPACE "USER\_TEMP" TEMPFILE '/u01/app/oracle/oradata/DBST670/user\_temp01.dbf' SIZE 16384K REUSE AUTOEXTEND ON NEXT 512K MAXSIZE 65536K EXTENT MANAGEMENT LOCAL;

TABLESPACE "USER\_TEMP" created.

SQL>

SQL> /\*Alter unprivileged user tablespace\*/

SQL> ALTER USER JKamando DEFAULT TABLESPACE user\_data1 TEMPORARY TABLESPACE user\_temp;

User JKAMANDO altered.

SQL> ALTER USER TOkusanya DEFAULT TABLESPACE user\_data1 TEMPORARY TABLESPACE user\_temp;

User TOKUSANYA altered.

SQL> ALTER USER JOluwadamilare DEFAULT TABLESPACE user\_data1 TEMPORARY TABLESPACE user\_temp;

User JOLUWADAMILARE altered.

SQL> ALTER USER BRussell DEFAULT TABLESPACE user\_data1 TEMPORARY TABLESPACE user\_temp;

User BRUSSELL altered.

SQL> ALTER USER RSeyoum DEFAULT TABLESPACE user\_data1 TEMPORARY TABLESPACE user\_temp;

User RSEYOUM altered.

SQL> ALTER USER DBAUser DEFAULT TABLESPACE user\_data1 TEMPORARY TABLESPACE user\_temp;

User DBAUSER altered.

SQL> ALTER USER DataDesignLeadUser DEFAULT TABLESPACE user\_data1 TEMPORARY TABLESPACE user\_temp;

User DATADESIGNLEADUSER altered.

SQL>

SQL> /\*Disconnect from DBAUser, Login as DataDesignLeadUser\*/

SQL> SELECT 'alter system kill session '''

2 || sid

3 || ','

4 || serial#

5 || ''';'

6 FROM v$session

7 WHERE username = 'DBAUSER';

'ALTERSYSTEMKILLSESSION'''||SID||','||SERIAL#||''';'

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

alter system kill session '268,10717';

SQL>

SQL> DISCONNECT;

Disconnected from Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production

SQL>

SQL> SHOW USER;

USER is ""

SQL>

SQL> CONNECT DataDesignLeadUser/abc123;

Connected.

SQL>

SQL> SHOW USER;

USER is "DATADESIGNLEADUSER"

## **Create Tables and Indexes**

SQL>

SQL> /\*---------------------------------------------------\*/

SQL> /\* CREATE TABLES \*/

SQL> /\*---------------------------------------------------\*/

SQL>

SQL> CREATE TABLE zip (

2 zipcode CHAR(10) NOT NULL,

3 city CHAR(25),

4 state CHAR(2),

5 CONSTRAINT pk\_zip PRIMARY KEY ( zipcode )

6 USING INDEX TABLESPACE user\_indx1

7 STORAGE ( INITIAL 1000 K NEXT 120 K )

8 )

9 PARTITION BY RANGE (

10 state

11 )

12 ( PARTITION zip\_part1

13 VALUES LESS THAN ( 'N' )

14 PCTFREE 5 PCTUSED 90 TABLESPACE user\_data1

15 STORAGE ( INITIAL 1092 k NEXT 200 k MAXEXTENTS UNLIMITED PCTINCREASE 0 ),

16 PARTITION zip\_part2

17 VALUES LESS THAN ( MAXVALUE )

18 PCTFREE 5 PCTUSED 90 TABLESPACE user\_data2

19 STORAGE ( INITIAL 1092 k NEXT 200 k MAXEXTENTS UNLIMITED PCTINCREASE 0 )

20 );

Table ZIP created.

SQL>

SQL> CREATE TABLE customer (

2 customerid NUMBER(10) NOT NULL,

3 firstname VARCHAR2(15),

4 lastname VARCHAR2(15),

5 phoneno CHAR(15),

6 address VARCHAR2(35),

7 zipcode CHAR(10),

8 CONSTRAINT pk\_customer\_id PRIMARY KEY ( customerid )

9 USING INDEX TABLESPACE user\_indx1

10 STORAGE ( INITIAL 50 K NEXT 10 K ),

11 CONSTRAINT fk\_zip FOREIGN KEY ( zipcode )

12 REFERENCES zip

13 )

14 PCTFREE 20 PCTUSED 40 TABLESPACE user\_data1

15 STORAGE ( INITIAL 120 k NEXT 10 k MAXEXTENTS UNLIMITED PCTINCREASE 0 );

Table CUSTOMER created.

SQL>

SQL> CREATE TABLE product (

2 productid NUMBER(10) NOT NULL,

3 prod\_unit\_price NUMBER(10),

4 description VARCHAR2(150),

5 CONSTRAINT pk\_productid PRIMARY KEY ( productid )

6 USING INDEX TABLESPACE user\_indx1

7 STORAGE ( INITIAL 50 K NEXT 10 K )

8 )

9 PCTFREE 5 PCTUSED 90 TABLESPACE user\_data2

10 STORAGE ( INITIAL 168 k NEXT 20 k MAXEXTENTS UNLIMITED PCTINCREASE 0 );

Table PRODUCT created.

SQL>

SQL> CREATE TABLE region (

2 regionid NUMBER(10) NOT NULL,

3 regionname VARCHAR(25),

4 description VARCHAR(150),

5 CONSTRAINT pk\_region PRIMARY KEY ( regionid )

6 USING INDEX TABLESPACE user\_indx1

7 STORAGE ( INITIAL 50 K NEXT 10 K )

8 )

9 PCTFREE 5 PCTUSED 90 TABLESPACE user\_data1

10 STORAGE ( INITIAL 192 k NEXT 20 k MAXEXTENTS UNLIMITED PCTINCREASE 0 );

Table REGION created.

SQL>

SQL> CREATE TABLE stores (

2 storeid NUMBER(10) NOT NULL,

3 regionid NUMBER(10),

4 storename VARCHAR2(15),

5 address VARCHAR2(35),

6 phoneno CHAR(15),

7 zipcode CHAR(10),

8 CONSTRAINT pk\_stores PRIMARY KEY ( storeid )

9 USING INDEX TABLESPACE user\_indx1

10 STORAGE ( INITIAL 50 K NEXT 10 K ),

11 CONSTRAINT fk\_store\_region FOREIGN KEY ( regionid )

12 REFERENCES region,

13 CONSTRAINT fk\_store\_zip FOREIGN KEY ( zipcode )

14 REFERENCES zip

15 )

16 PCTFREE 5 PCTUSED 90 TABLESPACE user\_data1

17 STORAGE ( INITIAL 96 k NEXT 10 k MAXEXTENTS UNLIMITED PCTINCREASE 0 );

Table STORES created.

SQL>

SQL> CREATE TABLE employee (

2 sales\_emp\_ssn NUMBER(9) NOT NULL,

3 storeid NUMBER(10),

4 firstname VARCHAR2(15),

5 lastname VARCHAR2(15),

6 address VARCHAR2(35),

7 phoneno CHAR(15),

8 zipcode CHAR(10),

9 CONSTRAINT pk\_sales\_emp\_ssn PRIMARY KEY ( sales\_emp\_ssn )

10 USING INDEX TABLESPACE user\_indx1

11 STORAGE ( INITIAL 50 K NEXT 10 K ),

12 CONSTRAINT fk\_emp\_storeid FOREIGN KEY ( storeid )

13 REFERENCES stores,

14 CONSTRAINT fk\_emp\_zip FOREIGN KEY ( zipcode )

15 REFERENCES zip

16 )

17 PCTFREE 20 PCTUSED 40 TABLESPACE user\_data1

18 STORAGE ( INITIAL 120 k NEXT 10 k MAXEXTENTS UNLIMITED PCTINCREASE 0 );

Table EMPLOYEE created.

SQL>

SQL> CREATE TABLE orders (

2 orderid NUMBER(10) NOT NULL,

3 customerid NUMBER(10) NOT NULL,

4 sales\_emp\_ssn NUMBER(9) NOT NULL,

5 order\_method CHAR(10),

6 CONSTRAINT pk\_orderid PRIMARY KEY ( orderid )

7 USING INDEX TABLESPACE user\_indx1

8 STORAGE ( INITIAL 50 K NEXT 10 K ),

9 CONSTRAINT fk\_customerid FOREIGN KEY ( customerid )

10 REFERENCES customer,

11 CONSTRAINT fk\_emp\_ssn FOREIGN KEY ( sales\_emp\_ssn )

12 REFERENCES employee

13 )

14 PCTFREE 5 PCTUSED 60 TABLESPACE user\_data1

15 STORAGE ( INITIAL 24 k NEXT 10 k MAXEXTENTS UNLIMITED PCTINCREASE 0 );

Table ORDERS created.

SQL>

SQL> CREATE TABLE order\_item (

2 orderid NUMBER(10) NOT NULL,

3 productid NUMBER(10),

4 quantity NUMBER(10),

5 CONSTRAINT pk\_order\_item PRIMARY KEY ( orderid,

6 productid )

7 USING INDEX TABLESPACE user\_indx1

8 STORAGE ( INITIAL 50 K NEXT 10 K ),

9 CONSTRAINT fk\_order\_id1 FOREIGN KEY ( orderid )

10 REFERENCES orders,

11 CONSTRAINT fk\_product\_id FOREIGN KEY ( productid )

12 REFERENCES product

13 )

14 PCTFREE 5 PCTUSED 90 TABLESPACE user\_data2

15 STORAGE ( INITIAL 16 k NEXT 20 k MAXEXTENTS UNLIMITED PCTINCREASE 0 );

Table ORDER\_ITEM created.

SQL>

SQL> CREATE TABLE payment (

2 paymentid NUMBER(10) NOT NULL,

3 orderid NUMBER(10),

4 card\_exp\_date DATE,

5 card\_number NUMBER(30),

6 payment\_amount NUMBER(20,2),

7 payment\_method VARCHAR2(5),

8 CONSTRAINT pk\_paymentid PRIMARY KEY ( paymentid )

9 USING INDEX TABLESPACE user\_indx1

10 STORAGE ( INITIAL 50 K NEXT 10 K ),

11 CONSTRAINT fk\_order\_id2 FOREIGN KEY ( orderid )

12 REFERENCES orders

13 )

14 PCTFREE 20 PCTUSED 40 TABLESPACE user\_data2

15 STORAGE ( INITIAL 40 k NEXT 10 k MAXEXTENTS UNLIMITED PCTINCREASE 0 );

Table PAYMENT created.

SQL>

SQL>

SQL> CREATE TABLE product\_cat (

2 prodcatid NUMBER(10) NOT NULL,

3 prodcattype VARCHAR(25),

4 description VARCHAR(150),

5 CONSTRAINT pk\_prodcat PRIMARY KEY ( prodcatid )

6 USING INDEX TABLESPACE user\_indx1

7 STORAGE ( INITIAL 50 K NEXT 10 K )

8 )

9 PCTFREE 5 PCTUSED 90 TABLESPACE user\_data1

10 STORAGE ( INITIAL 192 k NEXT 20 k MAXEXTENTS UNLIMITED PCTINCREASE 0 );

Table PRODUCT\_CAT created.

SQL>

SQL>

SQL> /\*Alter Product table to include additional Product Category ID\*/

SQL> ALTER TABLE PRODUCT ADD PRODCATID NUMBER (10)

2 CONSTRAINT fk\_prodcatid references PRODUCT\_CAT(PRODCATID);

Table PRODUCT altered.

SQL>

SQL>

SQL> /\*--------------------------------------------\*/

SQL> /\* INDEXES \*/

SQL> /\*--------------------------------------------\*/

SQL> CREATE INDEX fk\_zip ON

2 customer (

3 zipcode

4 )

5 TABLESPACE user\_indx1

6 STORAGE ( INITIAL 50 K NEXT 10 K );

Index FK\_ZIP created.

SQL>

SQL> CREATE INDEX fk\_customerid ON

2 orders (

3 customerid

4 )

5 TABLESPACE user\_indx1

6 STORAGE ( INITIAL 50 K NEXT 10 K );

Index FK\_CUSTOMERID created.

SQL>

SQL> CREATE INDEX fk\_order\_id1 ON

2 order\_item (

3 orderid

4 )

5 TABLESPACE user\_indx1

6 STORAGE ( INITIAL 50 K NEXT 10 K );

Index FK\_ORDER\_ID1 created.

SQL>

SQL> CREATE INDEX fk\_product\_id ON

2 order\_item (

3 productid

4 )

5 TABLESPACE user\_indx1

6 STORAGE ( INITIAL 50 K NEXT 10 K );

Index FK\_PRODUCT\_ID created.

SQL>

SQL> CREATE INDEX fk\_order\_id2 ON

2 payment (

3 orderid

4 )

5 TABLESPACE user\_indx1

6 STORAGE ( INITIAL 50 K NEXT 10 K );

Index FK\_ORDER\_ID2 created.

SQL>

SQL> CREATE INDEX fk\_emp\_storeid ON

2 employee (

3 storeid

4 )

5 TABLESPACE user\_indx1

6 STORAGE ( INITIAL 50 K NEXT 10 K );

Index FK\_EMP\_STOREID created.

SQL>

SQL> CREATE INDEX fk\_emp\_zip ON

2 employee (

3 zipcode

4 )

5 TABLESPACE user\_indx1

6 STORAGE ( INITIAL 50 K NEXT 10 K );

Index FK\_EMP\_ZIP created.

SQL>

SQL> CREATE INDEX fk\_store\_region ON

2 stores (

3 regionid

4 )

5 TABLESPACE user\_indx1

6 STORAGE ( INITIAL 50 K NEXT 10 K );

Index FK\_STORE\_REGION created.

SQL>

SQL> CREATE INDEX fk\_store\_zip ON

2 stores (

3 zipcode

4 )

5 TABLESPACE user\_indx1

6 STORAGE ( INITIAL 50 K NEXT 10 K );

Index FK\_STORE\_ZIP created.

SQL>

SQL> CREATE INDEX fk\_prodcatid ON

2 product (

3 prodcatid

4 )

5 TABLESPACE user\_indx1

6 STORAGE ( INITIAL 50 K NEXT 10 K );

Index FK\_PRODCATID created.

SQL>

SQL> CREATE INDEX fk\_emp\_ssn ON

2 orders (

3 sales\_emp\_ssn

4 )

5 TABLESPACE user\_indx1

6 STORAGE ( INITIAL 50 K NEXT 10 K );

Index FK\_EMP\_SSN created.

## **Load Data**

SQL>

SQL>

SQL> /\*Go to Console and Run the "RunSQLLoaders" file - Loads data into tables\*/

## **SQL Control File Loaders**

Customer.ctl

OPTIONS (SKIP=3, LOAD=1000)

LOAD

INFILE customer.dat

REPLACE INTO TABLE customer

FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'

(

CUSTOMERID

,FIRSTNAME

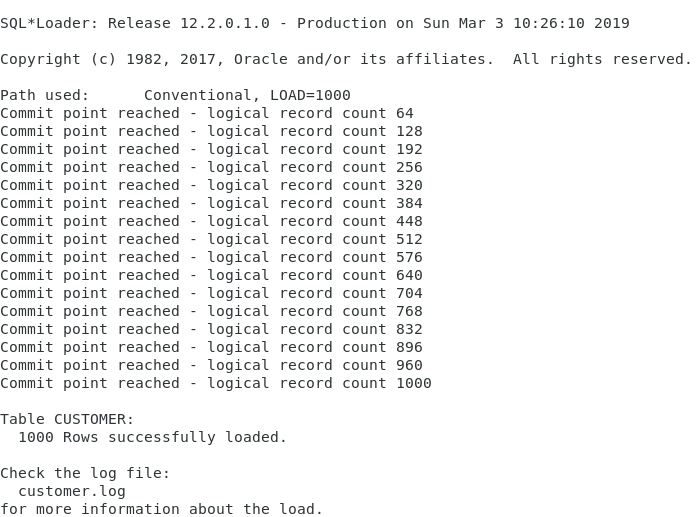
,LASTNAME

,PHONENO

,ADDRESS

,ZIPCODE

)



Employee.ctl

OPTIONS(LOAD=1000)

LOAD

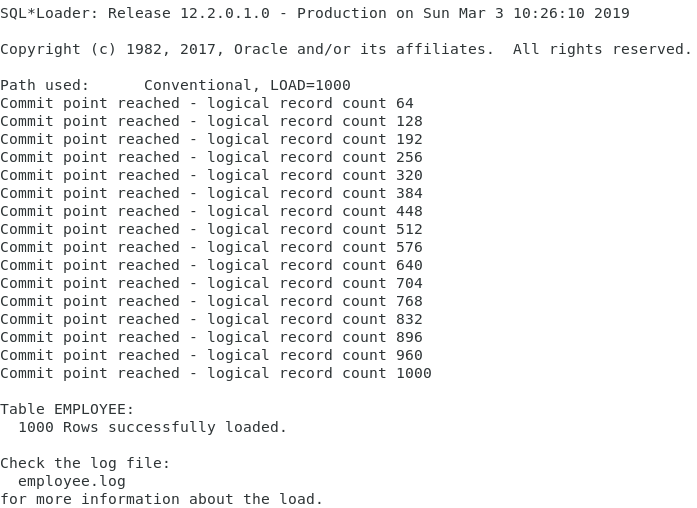
INFILE 'DBST670\_Employee.csv'

REPLACE

INTO TABLE employee

FIELDS TERMINATED BY ","

(SALES\_EMP\_SSN, STOREID,FIRSTNAME,LASTNAME, ADDRESS, PHONENO, ZIPCODE)



Order\_item.ctl

OPTIONS (SKIP=2, LOAD=1000)

LOAD DATA

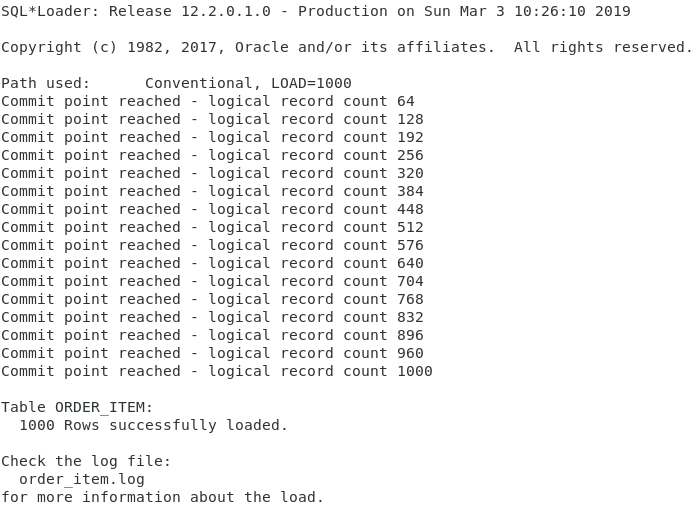
INFILE 'order\_item.dat'

INTO TABLE order\_item

(ORDERID TERMINATED BY WHITESPACE,

PRODUCTID TERMINATED BY WHITESPACE,

QUANTITY TERMINATED BY WHITESPACE)



Orders.ctl

OPTIONS (SKIP=3, LOAD=1000)

LOAD DATA

INFILE 'order.dat'

REPLACE

INTO TABLE orders

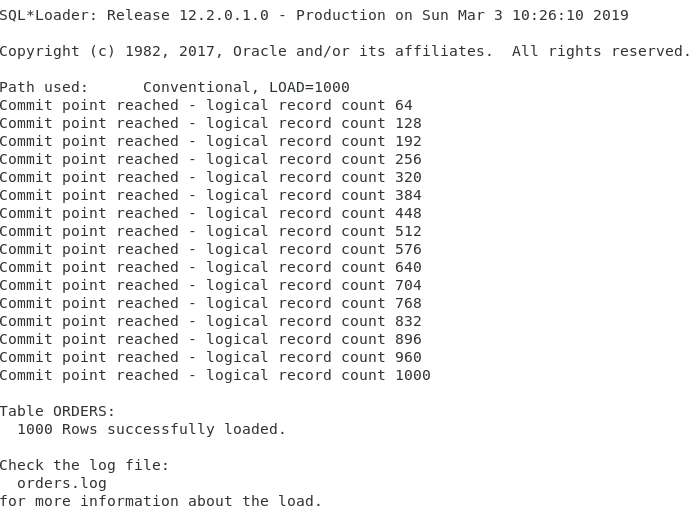
(ORDERID TERMINATED BY WHITESPACE,

CUSTOMERID TERMINATED BY WHITESPACE,

SALES\_EMP\_SSN TERMINATED BY WHITESPACE,

ORDER\_METHOD TERMINATED BY WHITESPACE

)



Payment.ctl

OPTIONS (SKIP=3, LOAD=1000)

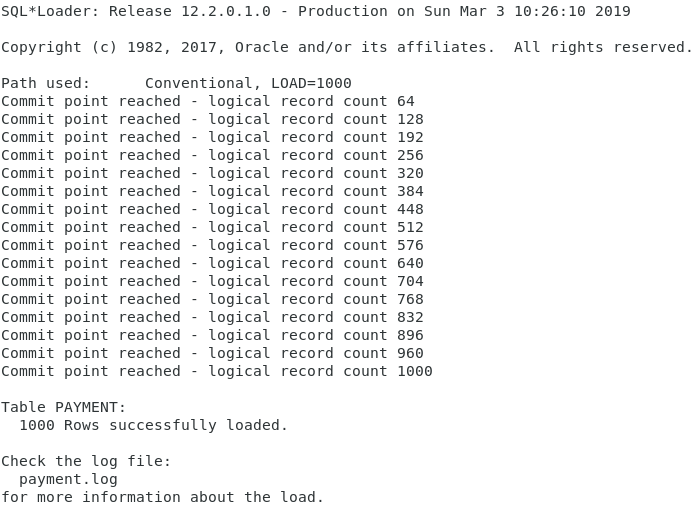
LOAD DATA

INFILE payment.dat

INTO TABLE payment

FIELDS TERMINATED BY ","

(PAYMENTID,ORDERID,CARD\_EXP\_DATE,CARD\_NUMBER,PAYMENT\_AMOUNT,PAYMENT\_METHOD)



Product.ctl

OPTIONS(SKIP=3, LOAD=1000)

LOAD

INFILE 'product.dat'

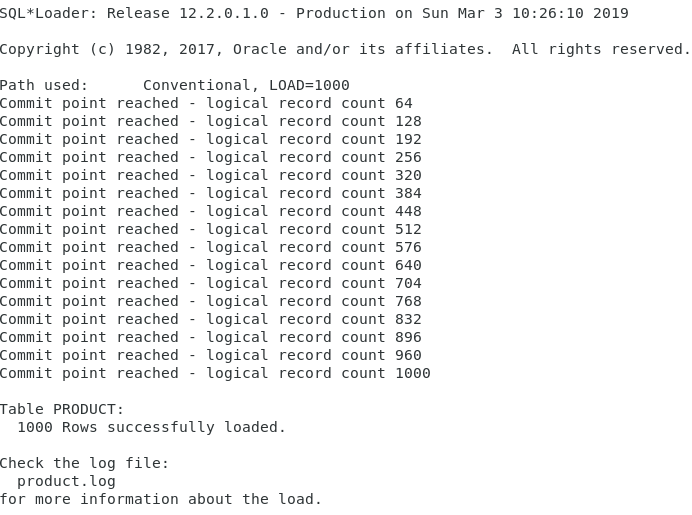
REPLACE

INTO TABLE product

(PRODUCTID TERMINATED BY WHITESPACE,

PROD\_UNIT\_PRICE TERMINATED BY WHITESPACE,

DESCRIPTION TERMINATED BY WHITESPACE)



Productcat.ctl

OPTIONS(LOAD=1000)

LOAD

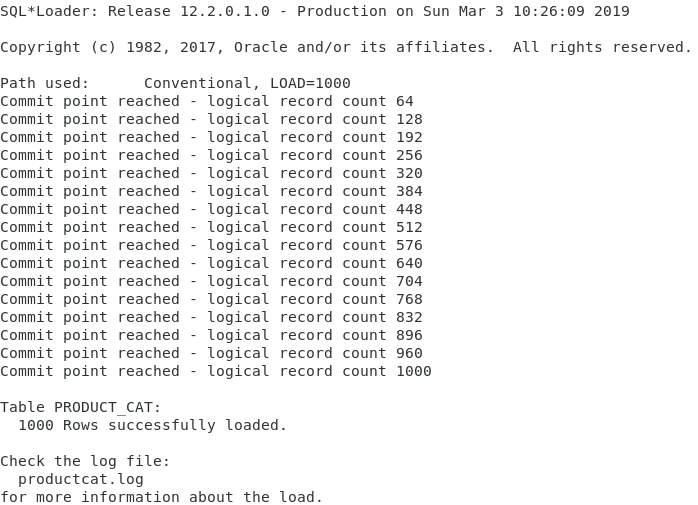
INFILE 'DBST670\_Product\_Cat.csv'

REPLACE

INTO TABLE product\_cat

FIELDS TERMINATED BY ","

(PRODCATID ,PRODCATTYPE, DESCRIPTION)



Region.ctl

OPTIONS(LOAD=1000)

LOAD

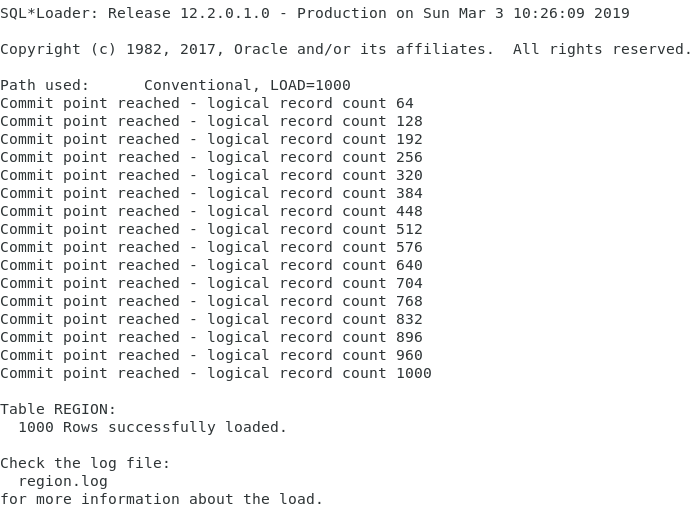
INFILE 'DBST670\_Region.csv'

REPLACE

INTO TABLE region

FIELDS TERMINATED BY ","

(REGIONID,REGIONNAME,DESCRIPTION)



Stores.ctl

OPTIONS(LOAD=1000)

LOAD

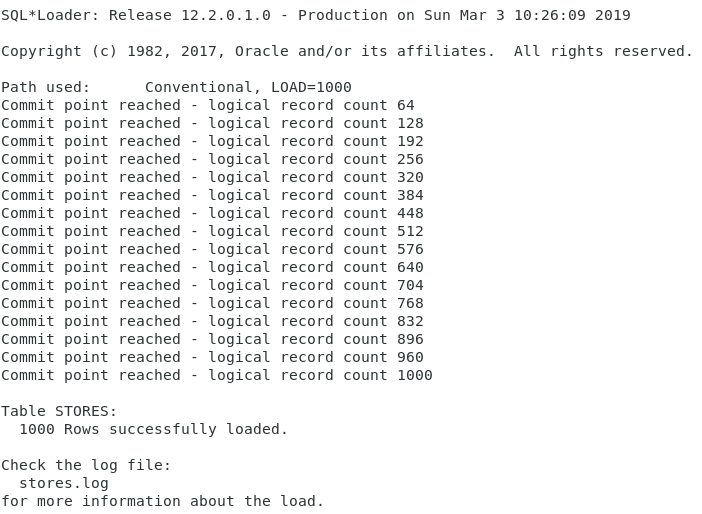
INFILE 'DBST670\_Store.csv'

REPLACE

INTO TABLE stores

FIELDS TERMINATED BY ","

(STOREID,REGIONID, STORENAME,ADDRESS, PHONENO, ZIPCODE)



Zip.ctl

OPTIONS (SKIP=2, LOAD=50000)

LOAD DATA

INFILE hb\_zip.dat

INTO TABLE zip

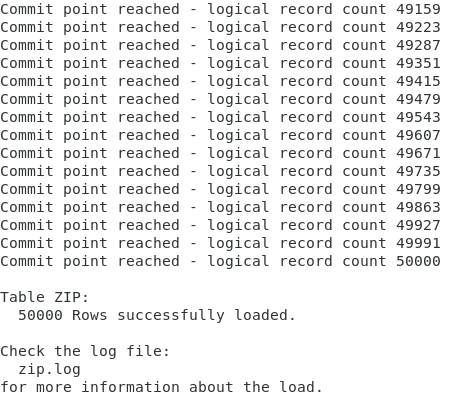
(

ZIPCODE POSITION(01:10) ,

CITY POSITION(11:25) ,

STATE POSITION(26:27)

)



SQL> /\*Fill gaps from Loader\*/

SQL> /\*Product table needs PRODCATID from Product\_Cat table\*/

SQL>

SQL> UPDATE product t1

2 SET

3 t1.prodcatid = (

4 SELECT prodcatid

5 FROM product\_cat t2

6 WHERE t2.prodcatid = t1.productid

7 )

8 WHERE EXISTS (

9 SELECT NULL

10 FROM product\_cat t2

11 WHERE t2.prodcatid = t1.productid

12 );

1,000 rows updated.

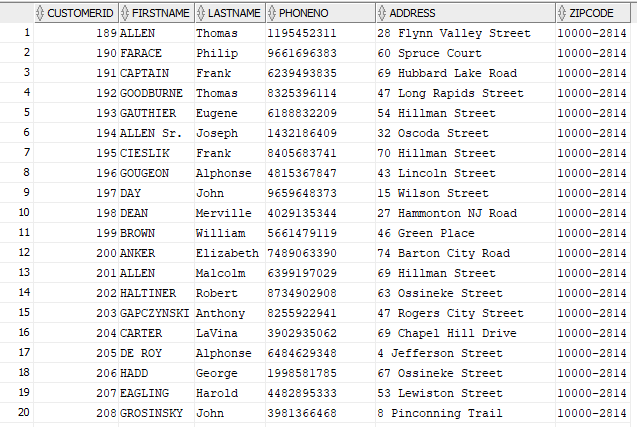
## **Query Tables**

SQL>

SQL> /\*Verify Data is loaded \*/

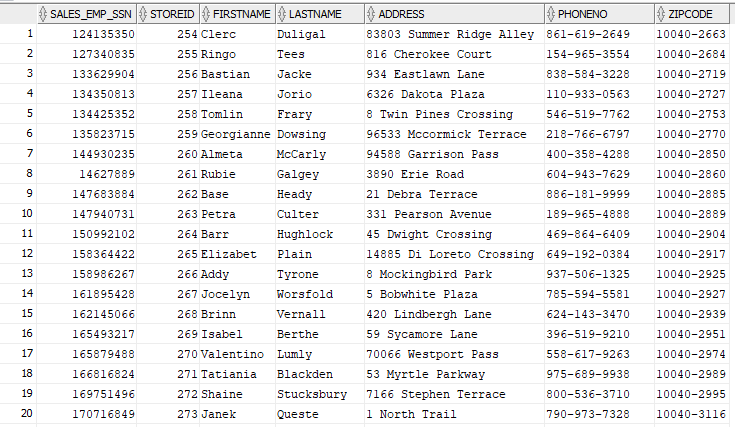
SQL>

SQL> SELECT \* FROM customer;



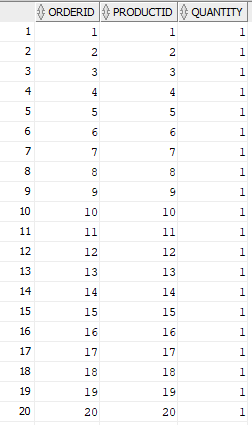
SQL>

SQL> SELECT \* FROM employee;



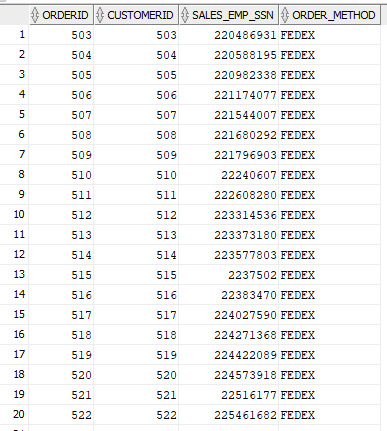
SQL>

SQL> SELECT \* FROM order\_item;



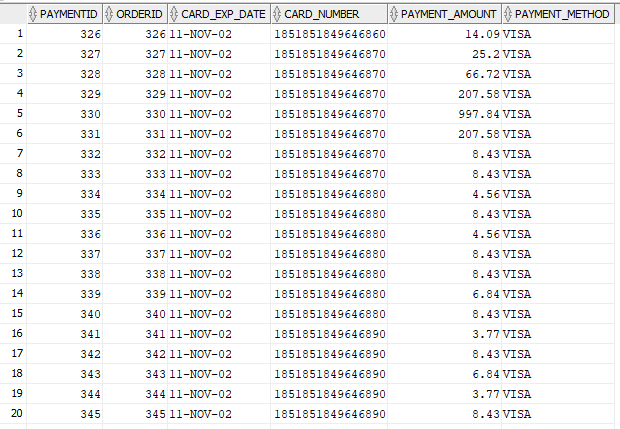
SQL>

SQL> SELECT \* FROM orders;



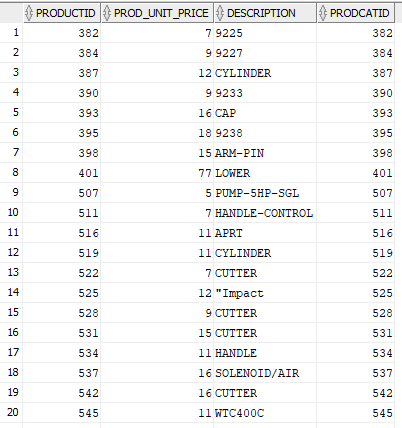
SQL>

SQL> SELECT \* FROM payment;



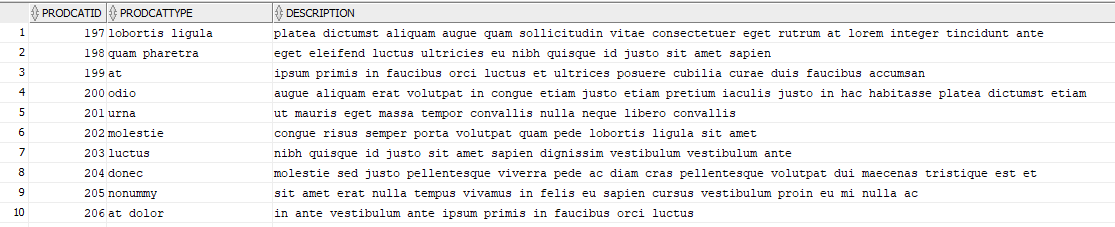
SQL>

SQL> SELECT \* FROM product;



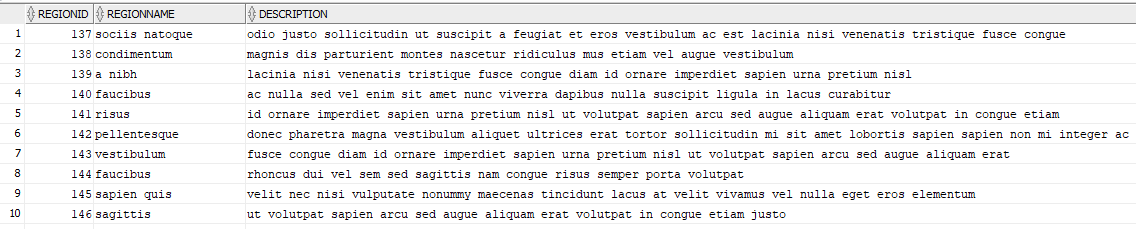
SQL>

SQL> SELECT \* FROM product\_cat;



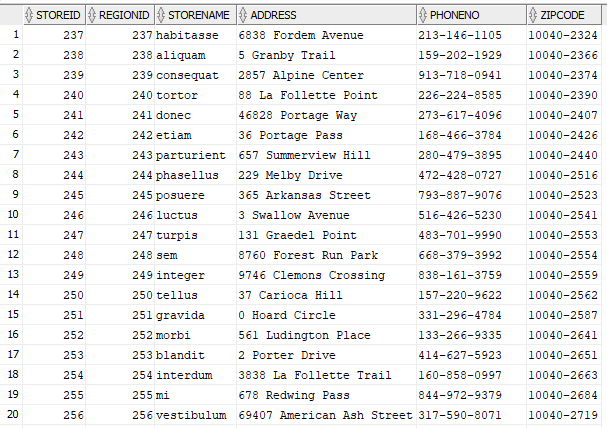
SQL>

SQL> SELECT \* FROM region;



SQL>

SQL> SELECT \* FROM stores;



SQL>

SQL> SELECT \* FROM zip;



# **Analyze Tables**

SQL>

SQL> /\*---------------------------------------------\*/

SQL> /\* ANALYZE TABLES \*/

SQL> /\*---------------------------------------------\*/

SQL>

SQL> /\*Analyze with deprecated command\*/

SQL>

SQL> ANALYZE TABLE customer COMPUTE STATISTICS;

Table CUSTOMER analyzed.

SQL> ANALYZE TABLE employee COMPUTE STATISTICS;

Table EMPLOYEE analyzed.

SQL> ANALYZE TABLE order\_item COMPUTE STATISTICS;

Table ORDER\_ITEM analyzed.

SQL> ANALYZE TABLE orders COMPUTE STATISTICS;

Table ORDERS analyzed.

SQL> ANALYZE TABLE payment COMPUTE STATISTICS;

Table PAYMENT analyzed.

SQL> ANALYZE TABLE product COMPUTE STATISTICS;

Table PRODUCT analyzed.

SQL> ANALYZE TABLE product\_cat COMPUTE STATISTICS;

Table PRODUCT\_CAT analyzed.

SQL> ANALYZE TABLE region COMPUTE STATISTICS;

Table REGION analyzed.

SQL> ANALYZE TABLE stores COMPUTE STATISTICS;

Table STORES analyzed.

SQL> ANALYZE TABLE zip COMPUTE STATISTICS;

Table ZIP analyzed.

SQL>

SQL> /\*Check results\*/

SQL>

SQL> SELECT num\_rows, blocks, empty\_blocks, avg\_space, avg\_row\_len

2 FROM user\_tables

3 WHERE table\_name ='CUSTOMER'

4 OR table\_name = 'EMPLOYEE'

5 OR table\_name = 'ORDER\_ITEM'

6 OR table\_name = 'ORDERS'

7 OR table\_name = 'PAYMENT'

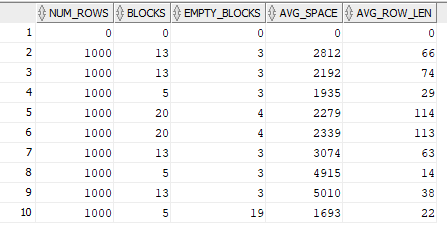
8 OR table\_name = 'PRODUCT'

9 OR table\_name = 'PRODUCT\_CAT'

10 OR table\_name = 'REGION'

11 OR table\_name = 'STORES'

12 OR table\_name = 'ZIP';



SQL>

SQL>

SQL> /\*Remove gathered statistics\*/

SQL>

SQL> ANALYZE TABLE customer DELETE STATISTICS;

Table CUSTOMER analyzed.

SQL> ANALYZE TABLE employee DELETE STATISTICS;

Table EMPLOYEE analyzed.

SQL> ANALYZE TABLE order\_item DELETE STATISTICS;

Table ORDER\_ITEM analyzed.

SQL> ANALYZE TABLE orders DELETE STATISTICS;

Table ORDERS analyzed.

SQL> ANALYZE TABLE payment DELETE STATISTICS;

Table PAYMENT analyzed.

SQL> ANALYZE TABLE product DELETE STATISTICS;

Table PRODUCT analyzed.

SQL> ANALYZE TABLE product\_cat DELETE STATISTICS;

Table PRODUCT\_CAT analyzed.

SQL> ANALYZE TABLE region DELETE STATISTICS;

Table REGION analyzed.

SQL> ANALYZE TABLE stores DELETE STATISTICS;

Table STORES analyzed.

SQL> ANALYZE TABLE zip DELETE STATISTICS;

Table ZIP analyzed.

SQL>

SQL> /\*Run DBMS\_STATS to get all stats\*/

SQL>

SQL> EXEC DBMS\_STATS.gather\_schema\_stats(ownname => 'DATADESIGNLEADUSER');

PL/SQL procedure successfully completed.

SQL>

SQL> /\*Check results\*/

SQL>

SQL> SELECT num\_rows, blocks, empty\_blocks, avg\_space, avg\_row\_len

2 FROM user\_tables

3 WHERE table\_name ='CUSTOMER'

4 OR table\_name = 'EMPLOYEE'

5 OR table\_name = 'ORDER\_ITEM'

6 OR table\_name = 'ORDERS'

7 OR table\_name = 'PAYMENT'

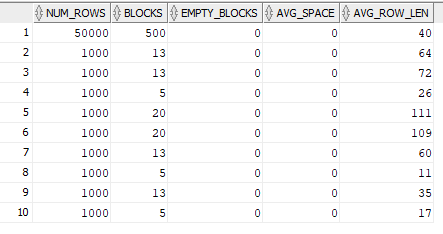
8 OR table\_name = 'PRODUCT'

9 OR table\_name = 'PRODUCT\_CAT'

10 OR table\_name = 'REGION'

11 OR table\_name = 'STORES'

12 OR table\_name = 'ZIP';



# **Check Object Status in Catalog**

SQL>

SQL> /\*---------------------------------------------\*/

SQL> /\* CHECK CONTENT OF CATALOG \*/

SQL> /\*---------------------------------------------\*/

SQL>

SQL> /\*Show valid status from all objects\*/

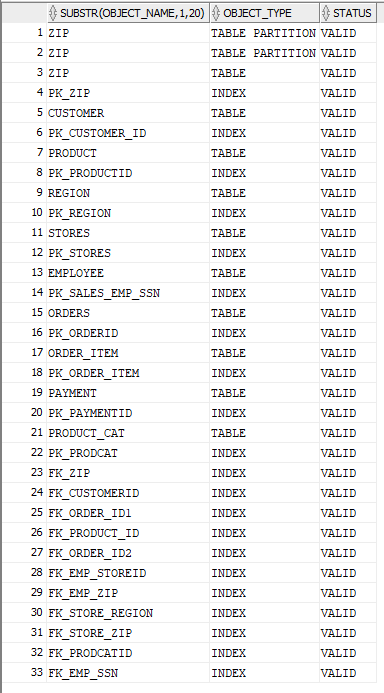
SQL> SELECT

2 substr

3 (object\_name,1,20),

4 object\_type,status FROM

5 user\_objects;



# **Query Row Numbers from Catalog**

SQL>

SQL> /\*Query row numbers from catalog\*/

SQL>

SQL> SELECT table\_name,

2 num\_rows,

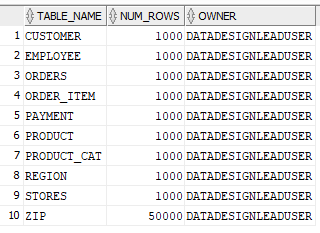
3 owner

4 FROM all\_tables

5 WHERE owner = 'DATADESIGNLEADUSER'

6 ORDER BY owner,

7 table\_name;



SQL>

SQL> COMMIT;

Commit complete.