

DBS311 Assignment #2

Instructions:

Part 1: Complete ERD with crow's foot notation based on Part 1 narrative. [5 marks]

Part 2: DDL/DML script to create the database and insert 5-10 records in each table. [10 marks]

Part 3: One stored procedure. [5 marks]

Part 4: A schema-less NoSQL design using sample data from part 2. You are to denote collection name with sample documents. [10 marks]

Submission: zip folder containing (1) a pdf file that will include the ERD and the screenshots of records and table structure (desc table name. select * from each of your tables) and the successful run of stored procedure with code. (2) SQL Script file containing the DDL statements to create tables with relationships and DML insert statements.

Part 1 BUSINESS NARRATIVE for “Virtual Songs Ltd.” music store

This case involves a Database creation for a Web-based music store called “Virtual Songs Ltd.” that sells music Discs over the Internet.

Virtual Songs Ltd. wants to keep information about their customers like: customer number, last name, first name, date of birth, street address, city, state (province), zip (postal code), phone and email address.

Each customer may order many orders (or even none if prospective client) and each order will contain: order number, order date and order total. Order must exist for only one customer. Customers will always pay for their order with the credit card, so information about it is stored in the dependent (on Order) listing called “Payment”. It will contain: card number, name on card, expiry date and card type (like Visa, Master, American-Express etc.). Order must be paid with the credit card before on-line transaction is saved and it has to be for at least one Disc. Every payment corresponds to an order and an order may contain multiple lines i.e., Discs.

There is listing of all Discs called “Recording” and it is explained through: recording id, title, performer's name, selling (or current) price and quantity in stock.

Each Disc belongs to a certain type or category of music like R&B, Jazz, Metal Rock, Alt Rock, Easy Rock, Classic Rock, Hip-Hop etc. and this will be stored in a look-up listing with category id and category description.

Also, each Disc is produced by a certain music label (company) like Sony, Columbia, Virgin etc and it is explained through: label id, label name and its URL site.

You should assume that there may exist a brand new category of music so that no Disc is of that type yet, but for the label you will assume it can not be without any Disc in the catalog.

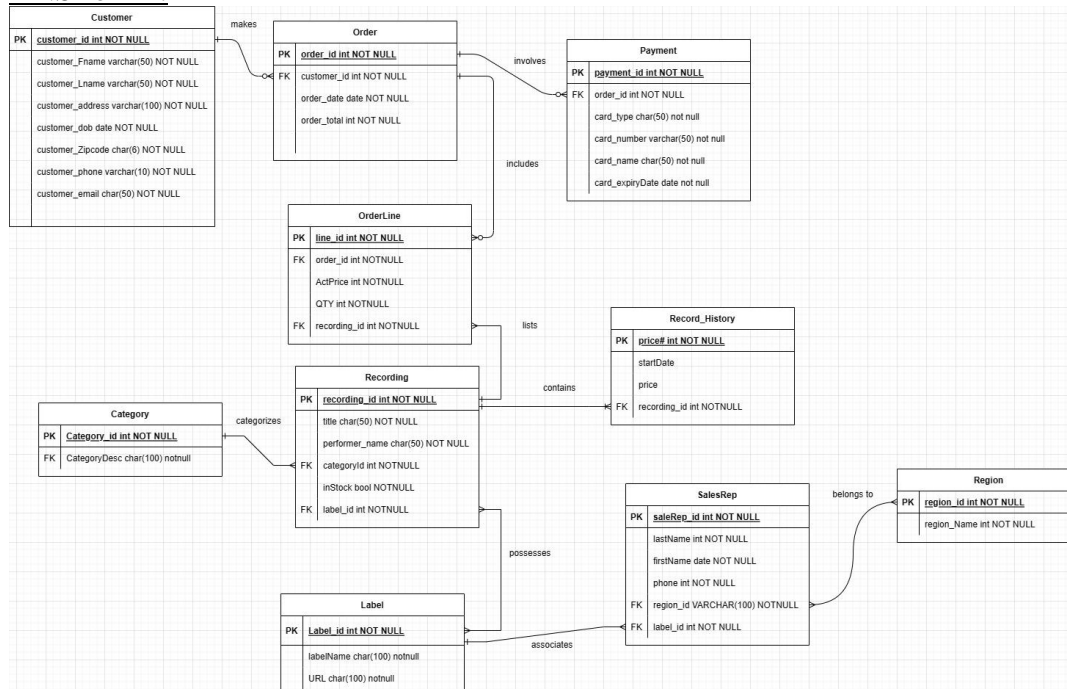
Each order line will contain information about the actual price and quantity ordered. This price is a historical price of a Disc at the time of being sold to the customer. For example, Disc called “Eyes Open” by “Snow Patrol” may have cost \$14 in September, but in November it may have cost \$16 and for the Christmas sale only \$12. The latest actual price is then equaled to the selling price in the “Recording” listing (this will be done through an Update Trigger in the Application code). So, selling or current price is always the most current price of the Disc.

Music store also wants to track a history of price changes for any Disc, so that dependent listing “Recording History” needs to be created where each Disc can be tracked by price number (like 1, 2, 3 etc.) its old price (like \$14, \$16, \$12) and the start date of each price like (09-SEP-06, 02-NOV-06 and 10-DEC-06).

“Virtual Tunes Ltd.” manages its Disc stock purchase by dealing with label’s sales reps. Each label will have one or more sales reps and they are tracked by: rep number, last name, first name, phone number and email address. Each sales rep has divided its territory in regions, where they keep info only about region id and region name (like US-North, US-West, Canada, Europe, Asia etc.) It is possible that same region is covered with none or more than one sales rep.

Note: Create meaningful names for all attributes and use # symbol for PK and (FK) for FK. Include ALL attributes for each entity incl. composite PK and every FK. Complete ERD should be in 2NF and must contain cardinality ratios and clear relationship notations.

ANS for #1:



Part 2 DDL/DML script

Using the Songs ERD, you need to provide SQL scripts with DDL / DML statements that will implement a Physical Solution for that Database Scenario as:

1. DDL scripts for creation of all Tables from the ERD.
2. Adding all PK, FK, UK and CK Constraints needed for these tables either with CREATE TABLE (UK, CK) or later with ALTER TABLE statements (PK, FK)
3. DML scripts to insert for at least five rows of sample data for each table

Check constraints you must implement:

- *OrderTotal* in table ORDERS must be a positive number
- *CardType* in table PAYMENT must belong to the set of values {VISA, MAST, AMEX}
- *CategoryDesc* must belong to the set of values {'R&B', 'Jazz', 'Metal Rock', 'Alt Rock', 'Easy Rock', 'Classic Rock', 'Hip-Hop'}
- *Qty* in table ORDER_LINE must be non-negative integer number
- *ActPrice* in table ORDER_LINE must be greater than zero
- *Price* in table RECORD_HIST must be positive decimal number not greater than \$999.99
- *You must add not null constraints as you see fit.*

Unique constraints (Not shown in ERD):

- *Email* in table CUSTOMER
- *Email* in table SALES_REP

ANS for #2 with typed code and screenshots

SET ECHO OFF;

DROP TABLE customer cascade constraints;

DROP TABLE orders cascade constraints;

DROP TABLE payment cascade constraints;

DROP TABLE categories cascade constraints;

DROP TABLE RepRegion cascade constraints;

DROP TABLE label cascade constraints;

DROP TABLE Recording cascade constraints;

DROP TABLE RecordingHistory cascade constraints;

DROP TABLE SalesRep cascade constraints;

DROP TABLE ORDERLINES cascade constraints;

/*

SET ECHO OFF;

--TABLE DROP STATEMENTS

SET VERIFICATION OFF

BEGIN

EXECUTE IMMEDIATE 'DROP TABLE customers CASCADE CONSTRAINTS';

EXCEPTION

WHEN OTHERS THEN NULL;

END;

/

BEGIN

EXECUTE IMMEDIATE 'DROP TABLE orders CASCADE CONSTRAINTS';

EXCEPTION

WHEN OTHERS THEN NULL;

END;

/

BEGIN

EXECUTE IMMEDIATE 'DROP TABLE payment CASCADE CONSTRAINTS';

EXCEPTION

WHEN OTHERS THEN NULL;

END;

/

BEGIN

EXECUTE IMMEDIATE 'DROP TABLE categories CASCADE CONSTRAINTS';

EXCEPTION

WHEN OTHERS THEN NULL;

END;

*/

-- Table customer (DDL)

--OK

PROMPT '***** Create customer table';

```
CREATE TABLE customer (  
    CID INT NOT NULL PRIMARY KEY,  
    CLastName varchar(50) NOT NULL,  
    CFirstName varchar(50) NOT NULL,  
    CDateBirth date NOT NULL,  
    CAddress varchar(50) NOT NULL,  
    Province varchar(50) DEFAULT NULL,  
    CpostalCode varchar(15) DEFAULT NULL,  
    phone varchar(50) NOT NULL,  
    Cemail varchar(50) NOT NULL,  
    UNIQUE(Cemail)  
);
```

--OK

-- Data for the table 'CUSTOMER' (DML)

INSERT ALL

 INTO customer VALUES (1, 'Manzon', 'Mary', to_date('1993-01-06','yyyy-mm-dd'),
 '100 Market Street', 'QC', '94080', '+1 650 219 4782', 'dmurphy@classicmodelcars.com')

 INTO customer VALUES (2, 'Merck', 'Bob', to_date('1997-07-09','yyyy-mm-dd'),
 '1550 Court Place', 'SK', '02107', '+1 215 837 0825', 'bmerck@classicmodelcars.com')

 INTO customer VALUES (3, 'Morgon', 'John', to_date('1993-12-06','yyyy-mm-dd'),
 '898 East 57th Street', 'MB', '10022', '+1 212 555 3000',
 'gbondur@classicmodelcars.com')

 INTO customer VALUES (4, 'Jennings', 'Luis', to_date('1995-03-08','yyyy-mm-dd'),
 '74 Jeffrey Street', 'ON', '75017', '+33 14 723 4404', 'ljennings@classicmodelcars.com')

 INTO customer VALUES (5, 'Amsterdam', 'Sam', to_date('1997-02-27','yyyy-mm-dd'),
 '43 Rue Place', 'QC', '75817', '+1 650 723 4782', 'samsterdam@classicmodelcars.com')

 INTO customer VALUES (6, 'Altuz', 'Murphy', to_date('1984-09-22','yyyy-mm-dd'),
 '100 Market Drive', 'SK', '94080', '+1 215 719 5782', 'maltuz@classicmodelcars.com')

 INTO customer VALUES (7, 'McKenna', 'Baden', to_date('1997-07-26','yyyy-mm-dd'),
 '1550 Court Avenue', 'SK', '02107', '+1 215 837 0825',

```
'bmckenna@classicmodelcars.com')
```

```
    INTO customer VALUES (8, 'Monzese', 'Jonas', to_date('1989-05-30','yyyy-mm-dd'),  
'500 East 49th Street', 'MB', '10022', '+1 269 555 3000',  
'jmonzese@classicmodelcars.com')
```

```
    INTO customer VALUES (9, 'Lincoln', 'Kevin', to_date('1977-10-07','yyyy-mm-dd'),  
'80 East Avenue', 'ON', '75017', '+33 14 723 9955', 'klincoln@classicmodelcars.com')
```

```
    INTO customer VALUES (10, 'Bavarian', 'Silva', to_date('1999-09-09','yyyy-mm-dd'),  
'80 Place Street', 'SK', '75817', '+33 14 733 4090', 'sbavarian@classicmodelcars.com')
```

```
    SELECT * FROM DUAL;  
COMMIT;
```

```
-- Table orders (DDL)
```

```
--OK
```

```
PROMPT '***** Create orders table';
```

```
CREATE TABLE orders (  
    orderID int NOT NULL PRIMARY KEY,  
    orderDate date NOT NULL,  
    cID int NOT NULL,  
    orderTotal DECIMAL(10,2),  
    CONSTRAINT ordelGReated check(orderTotal>0),  
    CONSTRAINT orders_cust_fk FOREIGN KEY (cId) REFERENCES customer (CID)  
);
```

```
--OK
```

```
-- Data for the table 'ORDERS' (DML)
```

```
INSERT ALL
```

```
    INTO orders VALUES (10100,to_date('2003-9-16','yyyy-mm-dd'),1,1000)
```

```
    INTO orders VALUES (10101,to_date('2003-3-19','yyyy-mm-dd'),2,2345)
```

```
    INTO orders VALUES (10102,to_date('2003-4-11','yyyy-mm-dd'),3,1234)
```

```
    INTO orders VALUES (10103,to_date('2003-6-29','yyyy-mm-dd'),4,21312)
```

```
    INTO orders VALUES (10104,to_date('2003-8-01','yyyy-mm-dd'),5,32422)
```

```
    INTO orders VALUES (10105,to_date('2003-2-26','yyyy-mm-dd'),6,234234)
```

```
    INTO orders VALUES (10106,to_date('2003-1-29','yyyy-mm-dd'),7,324423)
```

```
    INTO orders VALUES (10107,to_date('2003-8-27','yyyy-mm-dd'),8,54353)
```

```
INTO orders VALUES (10108,to_date('2003-12-29','yyyy-mm-dd'),9,432)
INTO orders VALUES (10109,to_date('2003-11-27','yyyy-mm-dd'),10,42312)
SELECT * FROM DUAL;
COMMIT;
```

-- Table payment (DDL)

--OK

PROMPT '***** Create payment table';

```
CREATE TABLE payment (
  cardNumber int NOT NULL,
  orderId int NOT NULL,
  cardHolder varchar(50),
  expDate date NOT NULL,
  cardType varchar(50)CHECK(cardType IN('VISA', 'MAST', 'AMEX')),
  PRIMARY KEY (cardNumber),
  CONSTRAINT payments_Orders FOREIGN KEY (orderId) REFERENCES orders
(orderiD)
);
```

--OK

-- Data for the table payment (DML)

INSERT ALL

```
  INTO payment VALUES(4887271275383781,10100,'Manzon Mary',to_date('2004-
10','yyyy-mm'),'VISA')
```

```
  INTO payment VALUES(4887271275383782,10101,'Merck Bob',to_date('2003-
06','yyyy-mm'),'VISA')
```

```
  INTO payment VALUES(4887271275383783,10102,'Morgan John',to_date('2004-
12','yyyy-mm'),'VISA')
```

```
  INTO payment VALUES(4887271275383784,10103,'Jennings Luis',to_date('2004-
12','yyyy-mm'),'VISA')
```

```
  INTO payment VALUES(4887271275383785,10104,'Amsterdam Sam',to_date('2003-
06','yyyy-mm'),'AMEX')
```

```
  INTO payment VALUES(4887271275383786,10105,'Altuz Murphy',to_date('2004-
08','yyyy-mm'),'MAST')
```

```
    INTO payment VALUES(4887271275383787,10106,'McKenna Baden',to_date('2003-05','yyyy-mm'),'MAST')
```

```
    INTO payment VALUES(4887271275383788,10107,'Monzese Jonas',to_date('2005-03','yyyy-mm'),'VISA')
```

```
    INTO payment VALUES(4887271275383789,10108,'Lincoln Kevin',to_date('2005-09','yyyy-mm'),'AMEX')
```

```
    INTO payment VALUES(4887271275383790,10109,'Bavarian Silva',to_date('2005-11','yyyy-mm'),'VISA')
```

```
    SELECT * FROM DUAL;  
COMMIT;
```

```
-- Table categories (DDL)
```

```
--OK
```

```
PROMPT '***** Create Categories table';
```

```
CREATE TABLE categories(  
    categoryId INT NOT NULL,  
    categoryDesc VARCHAR(50) NOT NULL,  
    PRIMARY KEY (categoryId),  
    CONSTRAINT categoryValue CHECK(categoryDesc IN('RB', 'Jazz', 'Metal Rock', 'Alt  
Rock', 'Easy Rock', 'Classic Rock', 'Hip-Hop'))  
);
```

```
--OK
```

```
-- Data for the table categories (DML)
```

```
INSERT ALL
```

```
    INTO categories VALUES(1,'RB')
```

```
    INTO categories VALUES(2,'Jazz')
```

```
    INTO categories VALUES(3,'Metal Rock')
```

```
    INTO categories VALUES(4,'Alt Rock')
```

```
    INTO categories VALUES(5,'Easy Rock')
```

```
    INTO categories VALUES(6,'Classic Rock')
```

```
    INTO categories VALUES(7,'Hip-Hop')
```

```
SELECT * FROM DUAL;
```



```
COMMIT;
```

```
-- Table RepRegion (DDL)
```

```
--OK
```

```
PROMPT '***** Create RepRegion table';
```

```
CREATE TABLE RepRegion(  
    RegionID INT NOT NULL,  
    RegionName VARCHAR(50) NOT NULL,  
    PRIMARY KEY (RegionID)  
);
```

```
--OK
```

```
-- Data for the table RepRegion (DML)
```

```
INSERT ALL  
    INTO RepRegion VALUES(1,'US-North')  
    INTO RepRegion VALUES(2,'US-West')  
    INTO RepRegion VALUES(3,'Canada')  
    INTO RepRegion VALUES(4,'Asia')  
    INTO RepRegion VALUES(5,'Europe')  
    INTO RepRegion VALUES(6,'SouthAmerica')  
    SELECT * FROM DUAL;  
COMMIT;
```

```
-- Table label (DDL)
```

```
--OK
```

```
PROMPT '***** Create label table';
```

```
CREATE TABLE label (  
    LabelID int not null,  
    LabelName varchar(50),  
    URL varchar(100),  
    PRIMARY KEY (LabelID)  
);
```

--OK

--Data for the table label (DML)

INSERT ALL

 INTO label VALUES(1,'Sony','https://www.sonymusic.com/labels/')

 INTO label VALUES(2,'Columbia','https://www.columbiarecords.com/')

 INTO label VALUES(3,'Virgin','https://www.virgin.com/virgin-companies/virgin-records')

 INTO label VALUES(4,'Artista Nashville','https://www.artistanashville.com/labels')

 INTO label VALUES(5,'Elektra Records','https://www.elektramusicgroup.com/')

SELECT * FROM DUAL;

COMMIT;

-- ORDERS Table Recording (DDL)

--OK

PROMPT '***** Create Recording table';

CREATE TABLE Recording(

 recordingID INT NOT NULL,

 Title VARCHAR(50),

 Artist VARCHAR(50),

 SellPrice DECIMAL(10,2),

 QtyInStock INT NOT NULL,

 categoryId INT,

 LabelID INT,

 PRIMARY KEY (recordingID),

 CONSTRAINT cate_recor FOREIGN KEY (categoryId) REFERENCES categories (categoryId),

 CONSTRAINT recor_label FOREIGN KEY (LabelID) REFERENCES label (LabelID)

);

--OK

--Data for the table Recording (DML)

INSERT ALL

INTO Recording VALUES(34,'Fire of Power','Mira Bracha',55.69,12,1,4)

INTO Recording VALUES(99,'Thunder of','Shreya Anahid',67.60,66,2,4)

INTO Recording VALUES(76,'Whispers of Choices','Naomi Telesphore',68.18,11,3,1)

INTO Recording VALUES(01,'Metalcore Destruction','Iris Rosamund',54.15,75,4,2)

INTO Recording VALUES(66,'Dream Secrets','Reba Katlyn',44.39,45,6,1)

INTO Recording VALUES(13,'In Whispers','Hildred Rien',76.98,43,5,5)

SELECT * FROM DUAL;

COMMIT;

--ORDERS Table RecordingHistory (DDL)

--OK

PROMPT '***** Create RECORDING HISTORY table';

CREATE TABLE RecordingHistory(

price DECIMAL(10,2) not null CHECK(price>0 AND price<999.99),

recordingID int not null,

price_history DECIMAL(10,2),

startDate Date not null,

CONSTRAINT recording_history FOREIGN KEY (recordingID) REFERENCES
Recording (recordingID)

);

--OK

--Data for the table RecordingHistory (DML)

INSERT ALL

INTO RecordingHistory VALUES(98.21, 34, 19, to_date('2022-12','yyyy-mm'))

INTO RecordingHistory VALUES(44.12, 13, 49, to_date('1999-11','yyyy-mm'))

INTO RecordingHistory VALUES(123.123, 66, 90, to_date('1978-11','yyyy-mm'))

INTO RecordingHistory VALUES(12.546, 99, 20, to_date('1955-06','yyyy-mm'))

INTO RecordingHistory VALUES(47.456, 01, 60, to_date('1987-03','yyyy-mm'))

SELECT * FROM DUAL;

COMMIT;

```

-- ORDERS Table SalesRep (DDL)
--OK
PROMPT '***** Create SalesRep table';
CREATE TABLE SalesRep (
    Rid int not null,
    RLName varchar(50) NOT NULL,
    RFName varchar(50) NOT NULL,
    phone varchar(50) NOT NULL,
    LabelID int not null,
    RregionID int NOT NULL,
    REmail varchar(50) NOT NULL,
    PRIMARY KEY (Rid),
    UNIQUE(REmail),
    CONSTRAINT label_id FOREIGN KEY (LabelID) REFERENCES label (LabelID),
    CONSTRAINT saleRegion FOREIGN KEY (RregionID) REFERENCES RepRegion
    (RregionID)
);

--OK
--Data for the table SalesRep (DML)
INSERT ALL
    INTO SalesRep VALUES(1, 'Abel', 'Reyes', '+1 311 377-6274', 1, 1,
    'gumpish@gmail.com')
    INTO SalesRep VALUES(2, 'India', 'Thomas', '+1 332 312-1234', 1, 2,
    'johnbob@aol.com')
    INTO SalesRep VALUES(4, 'Roosevelt', 'Brennan', '+1 123 365-7274', 4, 1,
    'kmiller@optonline.net')
    INTO SalesRep VALUES(5, 'Velasequz', 'Maria', '+1 999 334-9856', 2, 3,
    'pedwards@msn.com')
    INTO SalesRep VALUES(6, 'Connor', 'Squires', '+1 848 949-9999', 1, 4,
    'graham@msn.com')
    SELECT * FROM DUAL;
COMMIT;

```

```

-- ORDERS Table orderlines (DDL)
--OK
PROMPT '***** Create ORDERLINES table';
CREATE TABLE orderlines(
    Line int NOT NULL,
    orderId int NOT NULL,
    Qty int NOT NULL CHECK(Qty>=0),
    actPrice decimal(10,3) NOT NULL CHECK(actPrice>0),
    recordingID int not null,
    PRIMARY KEY (LINE,orderId),
    CONSTRAINT Order_lines FOREIGN KEY (orderId) REFERENCES orders
(orderID),
    CONSTRAINT Order_recording FOREIGN KEY (recordingID) REFERENCES
Recording (recordingID)
);

--OK
--Data for the table SalesRep (DML)
INSERT ALL
    INTO ORDERLINES VALUES(123, 10100, 9876, 191, 34)
    INTO ORDERLINES VALUES(198, 10101, 9, 998, 99)
    INTO ORDERLINES VALUES(12, 10102, 10, 428, 76)
    INTO ORDERLINES VALUES(111, 10103, 996, 729, 01)
    INTO ORDERLINES VALUES(765, 10104, 988, 765, 66)
    INTO ORDERLINES VALUES(989, 10105, 543, 23, 13)
SELECT * FROM DUAL;
COMMIT;
/*
SET VERIFICATION ON
SET ECHO ON
*/

```

Screenshot of code and output:

Worksheet	Query Builder
	<pre>SET ECHO OFF; DROP TABLE customer cascade constraints; DROP TABLE orders cascade constraints; DROP TABLE payment cascade constraints; DROP TABLE categories cascade constraints; DROP TABLE RepRegion cascade constraints; DROP TABLE label cascade constraints; DROP TABLE Recording cascade constraints; DROP TABLE RecordingHistory cascade constraints; DROP TABLE SalesRep cascade constraints; DROP TABLE ORDERLINES cascade constraints; /* SET ECHO OFF; --TABLE DROP STATEMENTS SET VERIFICATION OFF BEGIN EXECUTE IMMEDIATE 'DROP TABLE customers CASCADE CONSTRAINTS'; EXCEPTION WHEN OTHERS THEN NULL; END; / BEGIN EXECUTE IMMEDIATE 'DROP TABLE orders CASCADE CONSTRAINTS'; EXCEPTION WHEN OTHERS THEN NULL; END; / BEGIN EXECUTE IMMEDIATE 'DROP TABLE payment CASCADE CONSTRAINTS'; EXCEPTION WHEN OTHERS THEN NULL; END; / BEGIN EXECUTE IMMEDIATE 'DROP TABLE categories CASCADE CONSTRAINTS'; EXCEPTION WHEN OTHERS THEN NULL; END; */</pre>

Worksheet	Query Builder
	<pre>-- Table customer (DDL) --OK PROMPT '***** Create customer table'; CREATE TABLE customer (CID INT NOT NULL PRIMARY KEY, LastName varchar(50) NOT NULL, CFirstName varchar(50) NOT NULL, CDateBirth date NOT NULL, CAddress varchar(50) NOT NULL, CProvince varchar(50) DEFAULT NULL, CPostalCode varchar(15) DEFAULT NULL, CPhone varchar(50) NOT NULL, CEmail varchar(50) NOT NULL, UNIQUE(CEmail)); --OK -- Data for the table 'CUSTOMER' (DML) INSERT ALL INTO customer VALUES (1, 'Manson', 'Mary', to_date('1993-01-06','yyyy-mm-dd'), '100 Market Street', 'QC', '94080', '+1 650 219 4782', 'dmurphy@classicmodelcars.com') INTO customer VALUES (2, 'Merck', 'Bob', to_date('1997-07-09','yyyy-mm-dd'), '1550 Court Place', 'SK', '02107', '+1 215 837 0825', 'bmerck@classicmodelcars.com') INTO customer VALUES (3, 'Morgon', 'John', to_date('1993-12-06','yyyy-mm-dd'), '898 East 57th Street', 'MB', '10022', '+1 212 555 3000', 'jbonding@classicmodelcars.com') INTO customer VALUES (4, 'Jennings', 'Luis', to_date('1995-03-08','yyyy-mm-dd'), '74 Jeffrey Street', 'ON', '75017', '+33 14 723 4404', 'ljennings@classicmodelcars.com') INTO customer VALUES (5, 'Amsterdam', 'Sam', to_date('1997-02-27','yyyy-mm-dd'), '43 Rue Place', 'OC', '75817', '+1 650 723 4782', 'samsterdam@classicmodelcars.com') INTO customer VALUES (6, 'Altuz', 'Murphy', to_date('1984-09-22','yyyy-mm-dd'), '100 Market Drive', 'SK', '94080', '+1 215 719 5782', 'maltuz@classicmodelcars.com') INTO customer VALUES (7, 'McKenna', 'Baden', to_date('1997-07-26','yyyy-mm-dd'), '1550 Court Avenue', 'SK', '02107', '+1 215 837 0825', 'bmckenna@classicmodelcars.com') INTO customer VALUES (8, 'Monzese', 'Jonas', to_date('1989-05-30','yyyy-mm-dd'), '500 East 49th Street', 'MB', '10022', '+1 269 555 3000', 'jmonzese@classicmodelcars.com') INTO customer VALUES (9, 'Lincoln', 'Kevin', to_date('1977-10-07','yyyy-mm-dd'), '80 East Avenue', 'ON', '75017', '+33 14 723 9955', 'klincoln@classicmodelcars.com') INTO customer VALUES (10, 'Bavarian', 'Silva', to_date('1999-09-09','yyyy-mm-dd'), '80 Place Street', 'SK', '75817', '+33 14 733 4090', 'sbavarian@classicmodelcars.com') SELECT * FROM DUAL; COMMIT; -- Table orders (DDL) --OK PROMPT '***** Create orders table'; CREATE TABLE orders (orderID int NOT NULL PRIMARY KEY, orderDate date NOT NULL, CID int NOT NULL, orderTotal DECIMAL(10,2), CONSTRAINT ordelGreated check(orderTotal>0), CONSTRAINT orders_cust_fk FOREIGN KEY (CID) REFERENCES customer (CID));</pre>

```
Worksheet Query Builder
--OK
-- Data for the table 'ORDERS' (DML)
INSERT ALL
  INTO orders VALUES (10100,to_date('2003-9-16','yyyy-mm-dd'),1,1000)
  INTO orders VALUES (10101,to_date('2003-3-19','yyyy-mm-dd'),2,2345)
  INTO orders VALUES (10102,to_date('2003-4-11','yyyy-mm-dd'),3,1234)
  INTO orders VALUES (10103,to_date('2003-6-29','yyyy-mm-dd'),4,21312)
  INTO orders VALUES (10104,to_date('2003-8-01','yyyy-mm-dd'),5,32422)
  INTO orders VALUES (10105,to_date('2003-2-26','yyyy-mm-dd'),6,234234)
  INTO orders VALUES (10106,to_date('2003-1-29','yyyy-mm-dd'),7,324423)
  INTO orders VALUES (10107,to_date('2003-8-27','yyyy-mm-dd'),8,54353)
  INTO orders VALUES (10108,to_date('2003-12-29','yyyy-mm-dd'),9,432)
  INTO orders VALUES (10109,to_date('2003-11-27','yyyy-mm-dd'),10,42312)
SELECT * FROM DUAL;
COMMIT;

-- Table payment (DDL)
--OK
PROMPT '***** Create payment table';
CREATE TABLE payment (
  cardNumber int NOT NULL,
  orderId int NOT NULL,
  cardHolder varchar(50),
  expDate date NOT NULL,
  cardType varchar(50)CHECK(cardType IN('VISA', 'MAST', 'AMEX')),
  PRIMARY KEY (cardNumber),
  CONSTRAINT payments_Orders FOREIGN KEY (orderId) REFERENCES orders (orderId)
);

--OK
-- Data for the table payment (DML)
INSERT ALL
  INTO payment VALUES(4887271275383781,10100,'Manzon Mary',to_date('2004-10','yyyy-mm'),'VISA')
  INTO payment VALUES(4887271275383782,10101,'Merck Bob',to_date('2003-06','yyyy-mm'),'VISA')
  INTO payment VALUES(4887271275383783,10102,'Morgan John',to_date('2004-12','yyyy-mm'),'VISA')
  INTO payment VALUES(4887271275383784,10103,'Jennings Luis',to_date('2004-12','yyyy-mm'),'VISA')
  INTO payment VALUES(4887271275383785,10104,'Amsterdam Sam',to_date('2003-06','yyyy-mm'),'AMEX')
  INTO payment VALUES(4887271275383786,10105,'Altuz Murphy',to_date('2004-08','yyyy-mm'),'MAST')
  INTO payment VALUES(4887271275383787,10106,'McKenna Baden',to_date('2003-05','yyyy-mm'),'MAST')
  INTO payment VALUES(4887271275383788,10107,'Monzese Jonas',to_date('2005-03','yyyy-mm'),'VISA')
  INTO payment VALUES(4887271275383789,10108,'Lincoln Kevin',to_date('2005-09','yyyy-mm'),'AMEX')
  INTO payment VALUES(4887271275383790,10109,'Bavarian Silva',to_date('2005-11','yyyy-mm'),'VISA')
SELECT * FROM DUAL;
```

```
Worksheet Query Builder
-- COMMIT;

-- Table categories (DDL)
--OK
PROMPT '***** Create Categories table';
CREATE TABLE categories(
    categoryId INT NOT NULL,
    categoryDesc VARCHAR(50) NOT NULL,
    PRIMARY KEY (categoryId),
    CONSTRAINT categoryValue CHECK(categoryDesc IN('RB', 'Jazz', 'Metal Rock', 'Alt Rock', 'Easy Rock', 'Classic Rock', 'Hip-Hop'))
);

--OK
-- Data for the table categories (DML)
INSERT ALL
    INTO categories VALUES(1,'RB')
    INTO categories VALUES(2,'Jazz')
    INTO categories VALUES(3,'Metal Rock')
    INTO categories VALUES(4,'Alt Rock')
    INTO categories VALUES(5,'Easy Rock')
    INTO categories VALUES(6,'Classic Rock')
    INTO categories VALUES(7,'Hip-Hop')
SELECT * FROM DUAL;
COMMIT;

-- Table RepRegion (DDL)
--OK
PROMPT '***** Create RepRegion table';
CREATE TABLE RepRegion(
    RegionID INT NOT NULL,
    RegionName VARCHAR(50) NOT NULL,
    PRIMARY KEY (RegionID)
);

--OK
-- Data for the table RepRegion (DML)
INSERT ALL
    INTO RepRegion VALUES(1,'US-North')
    INTO RepRegion VALUES(2,'US-West')
    INTO RepRegion VALUES(3,'Canada')
    INTO RepRegion VALUES(4,'Asia')
    INTO RepRegion VALUES(5,'Europe')
    INTO RepRegion VALUES(6,'SouthAmerica')
SELECT * FROM DUAL;
```

```
Worksheet Query Builder
-- COMMIT;

-- Table label (DDL)
--OK
PROMPT '***** Create label table';
CREATE TABLE label (
    LabelID int not null,
    LabelName varchar(50),
    URL varchar(100),
    PRIMARY KEY (LabelID)
);

--OK
--Data for the table label (DML)
INSERT ALL
    INTO label VALUES(1,'Sony','https://www.sonymusic.com/labels/')
    INTO label VALUES(2,'Columbia','https://www.columbiarecords.com/')
    INTO label VALUES(3,'Virgin','https://www.virgin.com/virgin-companies/virgin-records')
    INTO label VALUES(4,'Artista Nashville','https://www.artistanashville.com/labels')
    INTO label VALUES(5,'Elektra Records','https://www.elektramusicgroup.com/')
SELECT * FROM DUAL;
COMMIT;

-- ORDERS Table Recording (DDL)
--OK
PROMPT '***** Create Recording table';
CREATE TABLE Recording(
    recordingID INT NOT NULL,
    Title VARCHAR(50),
    Artist VARCHAR(50),
    SellPrice DECIMAL(10,2),
    QtyInStock INT NOT NULL,
    categoryId INT,
    LabelID INT,
    PRIMARY KEY (recordingID),
    CONSTRAINT cate_recor FOREIGN KEY (categoryId) REFERENCES categories (categoryId),
    CONSTRAINT recor_label FOREIGN KEY (LabelID) REFERENCES label (LabelID)
);
```



```
Worksheet Query Builder
--OK
--Data for the table Recording (DML)
INSERT ALL
  INTO Recording VALUES(34,'Fire of Power','Mira Bracha',55.69,12,1,4)
  INTO Recording VALUES(99,'Thunder of','Shreya Anahid',67.60,66,2,4)
  INTO Recording VALUES(76,'Whispers of Choices','Naomi Telesphore',68.18,11,3,1)
  INTO Recording VALUES(01,'Metalcore Destruction','Iris Rosamund',54.15,75,4,2)
  INTO Recording VALUES(66,'Dream Secrets','Reba Katlyn',44.39,45,6,1)
  INTO Recording VALUES(13,'In Whispers','Hildred Rien',76.98,43,5,5)
SELECT * FROM DUAL;
COMMIT;

--ORDERS Table RecordingHistory (DDL)
--OK
PROMPT '***** Create RECORDING HISTORY table';
CREATE TABLE RecordingHistory(
  price DECIMAL(10,2) not null CHECK(price>0 AND price<999.99),
  recordingID int not null,
  price_history DECIMAL(10,2),
  startDate Date not null,
  CONSTRAINT recording_history FOREIGN KEY (recordingID) REFERENCES Recording (recordingID)
);

--OK
--Data for the table RecordingHistory (DML)
INSERT ALL
  INTO RecordingHistory VALUES(98.21, 34, 19, to_date('2022-12','yyyy-mm'))
  INTO RecordingHistory VALUES(44.12, 13, 49, to_date('1999-11','yyyy-mm'))
  INTO RecordingHistory VALUES(123.123, 66, 90, to_date('1978-11','yyyy-mm'))
  INTO RecordingHistory VALUES(12.546, 99, 20, to_date('1955-06','yyyy-mm'))
  INTO RecordingHistory VALUES(47.456, 01, 60, to_date('1987-03','yyyy-mm'))
SELECT * FROM DUAL;
COMMIT;

-- ORDERS Table SalesRep (DDL)
--OK
PROMPT '***** Create SalesRep table';
CREATE TABLE SalesRep (
  Rid int not null,
  RLName varchar(50) NOT NULL,
  RFName varchar(50) NOT NULL,
  phone varchar(50) NOT NULL,
  LabelID int not null,
```

```
Worksheet Query Builder
  RregionID int NOT NULL,
  REmail varchar(50) NOT NULL,
  PRIMARY KEY (Rid),
  UNIQUE(REmail),
  CONSTRAINT label_id FOREIGN KEY (LabelID) REFERENCES label (LabelID),
  CONSTRAINT saleRegion FOREIGN KEY (RregionID) REFERENCES RepRegion (RregionID)
);

--OK
--Data for the table SalesRep (DML)
INSERT ALL
  INTO SalesRep VALUES(1, 'Abel', 'Reyes', '+1 311 377-6274', 1, 1, 'gumpish@gmail.com')
  INTO SalesRep VALUES(2, 'India', 'Thomas', '+1 332 312-1234', 1, 2, 'johnbob@aol.com')
  INTO SalesRep VALUES(4, 'Roosevelt', 'Brennan', '+1 123 365-7274', 4, 1, 'kmiller@optonline.net')
  INTO SalesRep VALUES(5, 'Velasequz', 'Maria', '+1 999 334-9856', 2, 3, 'pedwards@msn.com')
  INTO SalesRep VALUES(6, 'Connor', 'Squires', '+1 848 949-9999', 1, 4, 'graham@msn.com')
SELECT * FROM DUAL;
COMMIT;

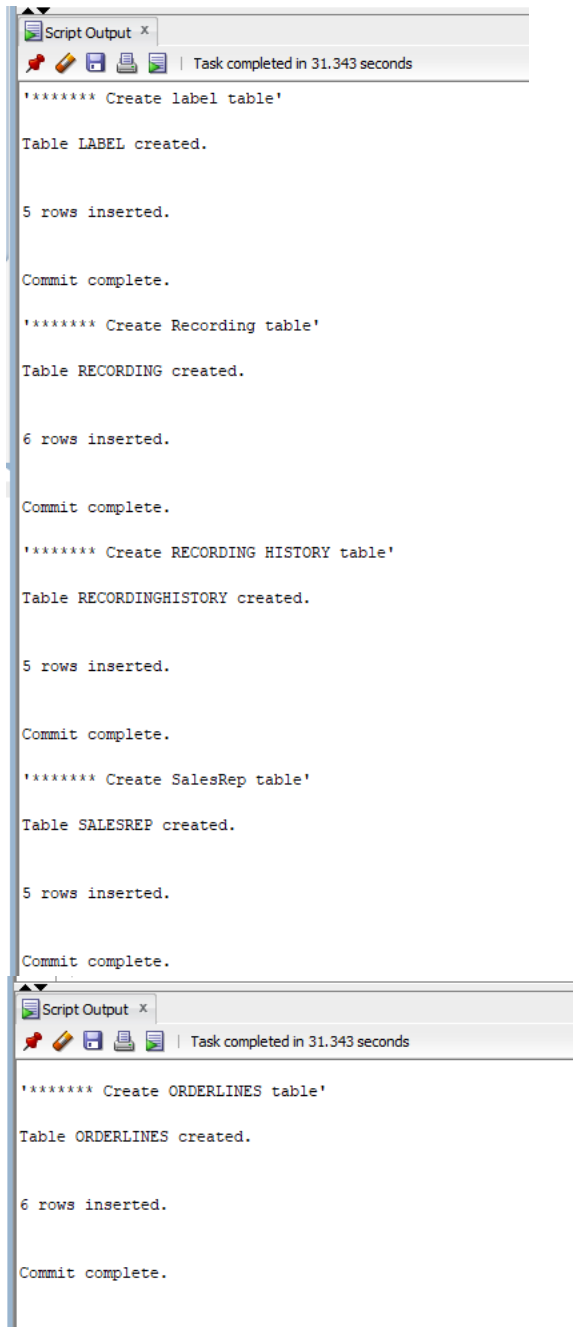
-- ORDERS Table orderlines (DDL)
--OK
PROMPT '***** Create ORDERLINES table';
CREATE TABLE orderlines(
  Line int NOT NULL,
  orderId int NOT NULL,
  Qty int NOT NULL CHECK(Qty>=0),
  actPrice decimal(10,3) NOT NULL CHECK(actPrice>0),
  recordingID int not null,
  PRIMARY KEY (LINE,orderId),
  CONSTRAINT Order_lines FOREIGN KEY (orderId) REFERENCES orders (orderId),
  CONSTRAINT Order_recording FOREIGN KEY (recordingID) REFERENCES Recording (recordingID)
);

--OK
--Data for the table SalesRep (DML)
INSERT ALL
  INTO ORDERLINES VALUES(123, 10100, 9976, 191, 34)
  INTO ORDERLINES VALUES(198, 10101, 9, 998, 99)
  INTO ORDERLINES VALUES(12, 10102, 10, 428, 76)
  INTO ORDERLINES VALUES(111, 10103, 996, 729, 01)
  INTO ORDERLINES VALUES(765, 10104, 988, 765, 66)
  INTO ORDERLINES VALUES(989, 10105, 543, 23, 13)
SELECT * FROM DUAL;
COMMIT;

/*
SET VERIFICATION ON
SET ECHO ON
*/

Script Output x
Task completed in 31.343 seconds
```

Script Output x Task completed in 31.343 seconds	Script Output x Task completed in 31.343 seconds
<p>Table CUSTOMER dropped.</p> <p>Table ORDERS dropped.</p> <p>Table PAYMENT dropped.</p> <p>Table CATEGORIES dropped.</p> <p>Table REPREGION dropped.</p> <p>Table LABEL dropped.</p> <p>Table RECORDING dropped.</p> <p>Table RECORDINGHISTORY dropped.</p> <p>Table SALESREP dropped.</p> <p>Table ORDERLINES dropped.</p> <p>***** Create customer table'</p> <p>Table CUSTOMER created.</p> <p>10 rows inserted.</p> <p>Commit complete.</p>	<p>***** Create orders table'</p> <p>Table ORDERS created.</p> <p>10 rows inserted.</p> <p>Commit complete.</p> <p>***** Create payment table'</p> <p>Table PAYMENT created.</p> <p>10 rows inserted.</p> <p>Commit complete.</p> <p>***** Create Categories table'</p> <p>Table CATEGORIES created.</p> <p>7 rows inserted.</p> <p>Commit complete.</p> <p>***** Create RepRegion table'</p> <p>Table REPREGION created.</p> <p>6 rows inserted.</p> <p>Commit complete.</p>



```
Script Output x
Task completed in 31.343 seconds

***** Create label table'

Table LABEL created.

5 rows inserted.

Commit complete.

***** Create Recording table'

Table RECORDING created.

6 rows inserted.

Commit complete.

***** Create RECORDING HISTORY table'

Table RECORDINGHISTORY created.

5 rows inserted.

Commit complete.

***** Create SalesRep table'

Table SALESREP created.

5 rows inserted.

Commit complete.

***** Create ORDERLINES table'

Table ORDERLINES created.

6 rows inserted.

Commit complete.
```

Part 3 Stored Procedure

Write a procedure named *Recording_Selling_Price_Report* to show the number of recordings in each price category as follows:

If selling price is less than the average selling price then the category is LOW
Else the category is HIGH

The stored procedure must print how many of LOW and HIGH recordings are available in the database.

ANS: Code and output for #3

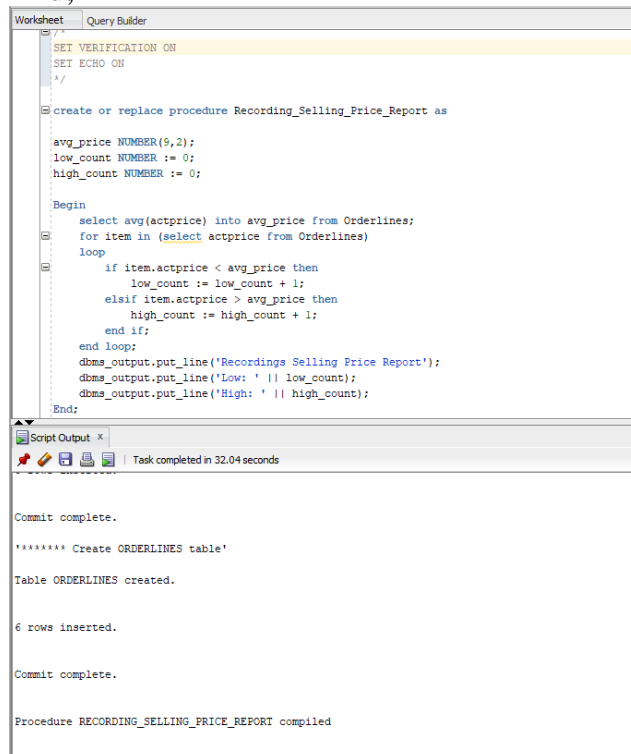
create or replace procedure Recording_Selling_Price_Report as

```
avg_price NUMBER(9,2);  
low_count NUMBER := 0;  
high_count NUMBER := 0;
```

Begin

```
select avg(actprice) into avg_price from Orderlines;  
for item in (select actprice from Orderlines)  
loop  
    if item.actprice < avg_price then  
        low_count := low_count + 1;  
    elsif item.actprice > avg_price then  
        high_count := high_count + 1;  
    end if;  
end loop;  
dbms_output.put_line('Recordings Selling Price Report');  
dbms_output.put_line('Low: ' || low_count);  
dbms_output.put_line('High: ' || high_count);
```

End;



The screenshot displays the SQL Developer interface. The top pane, titled 'Worksheet', shows the PL/SQL code for the 'Recording_Selling_Price_Report' procedure. The code includes variable declarations for 'avg_price', 'low_count', and 'high_count', followed by a 'Begin' block containing a query to calculate the average actprice, a loop to iterate through 'Orderlines' and count items below and above the average, and finally, output statements using 'dbms_output.put_line'. The bottom pane, titled 'Script Output', shows the execution results: 'Commit complete.', '***** Create ORDERLINES table', 'Table ORDERLINES created.', '6 rows inserted.', 'Commit complete.', and 'Procedure RECORDING_SELLING_PRICE_REPORT compiled'.

```
SET VERIFICATION ON  
SET ECHO ON  
*/  
  
create or replace procedure Recording_Selling_Price_Report as  
  
    avg_price NUMBER(9,2);  
    low_count NUMBER := 0;  
    high_count NUMBER := 0;  
  
    Begin  
        select avg(actprice) into avg_price from Orderlines;  
        for item in (select actprice from Orderlines)  
        loop  
            if item.actprice < avg_price then  
                low_count := low_count + 1;  
            elsif item.actprice > avg_price then  
                high_count := high_count + 1;  
            end if;  
        end loop;  
        dbms_output.put_line('Recordings Selling Price Report');  
        dbms_output.put_line('Low: ' || low_count);  
        dbms_output.put_line('High: ' || high_count);  
    End;
```

Script Output x
Task completed in 32.04 seconds

Commit complete.
***** Create ORDERLINES table'
Table ORDERLINES created.
6 rows inserted.
Commit complete.
Procedure RECORDING_SELLING_PRICE_REPORT compiled

Part 4 NoSQL Design

You need to identify the collections you need in this part and display sample json data inside them. For example, the Customer collection could contain

```
{“_id”:1,”LastName”:”Alam”,  
“FirstName”:”Tanvir”,”DoB”:”1/1/2000”,”Email”:”Tanvir.alam@senecacollege.ca”,  
”phone”:”800-800-8080”, “Zip”:”A1A1A1”} etc. You must use the sample data  
you used in Part 2.
```

You do not need to implement using MongoDB. However, you need to research how to handle 1-M relationship in NoSQL to display in your sample data design.

You can embed id in the order(check recording). You need to know one-to many relationship for MongoDB.

ANS for #4:

//CUSTOMER

```
{"id" : 1, "Lname" : "Abel", "Fname" : "Reyes", "phone" : "+1 311 377-6274",  
"lable_id" : 1, "region_id" : 1, "email" : "gumpish@gmail.com"}  
{"id" : 2, "Lname" : "India", "Fname" : "Thomas", "phone" : "+1 332 312-1234",  
"lable_id" : 1, "region_id" : 2, "email" : "johnbob@aol.com"}  
{"id" : 4, "Lname" : "Roosevelt", "Fname" : "Brennan", "phone" : "+1 123 365-  
7274", "lable_id" : 4, "region_id" : 1, "email" : "kmiller@optonline.net"}  
{"id" : 5, "Lname" : "Velasequz", "Fname" : "Maria", "phone" : "+1 999 334-  
9856", "lable_id" : 2, "region_id" : 3, "email" : "pedwards@msn.com"}  
{"id" : 6, "Lname" : "Connor", "Fname" : "Squires", "phone" : "+1 848 949-9999",  
"lable_id" : 1, "region_id" : 4, "email" : "graham@msn.com"}
```

//ORDER

```
{"id" : 10100, "date" : "2003-9-16", "customer_id" : 1, "total" : 1000}  
{"id" : 10101, "date" : "2003-3-19", "customer_id" : 2, "total" : 2345}  
{"id" : 10103, "date" : "2003-6-29", "customer_id" : 4, "total" : 21312}  
{"id" : 10104, "date" : "2003-8-01", "customer_id" : 5, "total" : 32422}  
{"id" : 10105, "date" : "2003-2-26", "customer_id" : 6, "total" : 234234}
```

//PAYMENT

```
{"Card_number":4887271275383781,"id":10100,"card_holder":"Manzon Mary","exp_date":"2004-10","card_type":"VISA"}
{"Card_number":4887271275383782,"id":10101,"card_holder":"Merck Bob","exp_date":"2003-06","card_type":"VISA"}
{"Card_number":4887271275383783,"id":10102,"card_holder":"Morgan John","exp_date":"2004-12","card_type":"VISA"}
{"Card_number":4887271275383784,"id":10103,"card_holder":"Jennings Luis","exp_date":"2004-12","card_type":"VISA"}
{"Card_number":4887271275383785,"id":10104,"card_holder":"Amsterdam Sam","exp_date":"2003-06","card_type":"AMEX"}
```

//RECORDING

```
{"id":34,"title":"Fire of Power","artist":"Mira Bracha","price":55.69,"stock":12,"category_id":1,"lable_id":4}
{"id":99,"title":"Thunder of","artist":"Shreya Anahid","price":67.60,"stock":66,"category_id":2,"lable_id":4}
{"id":76,"title":"Whispers of Choices","artist":"Naomi Telesphore","price":68.18,"stock":11,"category_id":3,"lable_id":1}
{"id":01,"title":"Metalcore Destruction","artist":"Iris Rosamund","price":54.15,"stock":75,"category_id":4,"lable_id":2}
{"id":66,"title":"Dream Secrets","artist":"Reba Katlyn","price":44.39,"stock":45,"category_id":6,"lable_id":1}
{"id":13,"title":"In Whispers","artist":"Hildred Rien","price":76.98,"stock":43,"category_id":5,"lable_id":5}
```

//RECORDING HISTORY

```
{"price":98.21,"recording_id":34,"price_history":19,"startd_date":"2022-12"}
{"price":44.12,"recording_id":13,"price_history":49,"startd_date":"1999-11"}
{"price":123.123,"recording_id":66,"price_history":90,"startd_date":"1978-11"}
{"price":12.546,"recording_id":99,"price_history":20,"startd_date":"1955-06"}
{"price":47.456,"recording_id":01,"price_history":60,"startd_date":"1987-03"}
```

// ORDERLINE

```
{"Line":123,"order_id":10100,"qty":9876,"act_price":191,"recording_id":34}
{"Line":198,"order_id":10101,"qty":9,"act_price":998,"recording_id":99}
{"Line":12,"order_id":10102,"qty":10,"act_price":428,"recording_id":76}
{"Line":765,"order_id":10104,"qty":988,"act_price":765,"recording_id":66}
```

```
{"Line":989,"order_id":10105,"qty":543,"act_price":23,"recording_id":13}
```

//SALESREP

```
{"id":1,"Abel","Reyes","+1 311 377-6274",1,1,"gumpish@gmail.com"}  
{"id":2,"India","Thomas","+1 332 312-1234",1,2,"johnbob@aol.com"}  
{"id":4,"Roosevelt","Brennan","+1 123 365-7274",4,1,  
"kmiller@optonline.net"}  
{"id":5,"Velasequz","Maria","+1 999 334-9856",2,3,"pedwards@msn.com"}  
{"id":6,"Connor","Squires","+1 848 949-9999",1,4,"graham@msn.com"}
```

//LABEL

```
{"id":1,"name ":"Sony","URL ": "https://www.sonymusic.com/labels/"}  
{"id":2,"name ":"Columnia","URL ": "https://www.columbiarecords.com/"}  
{"id":3,"name ":"Virgin","URL ": "https://www.virgin.com/virgin-  
companies/virgin-records"}  
{"id":4,"name ":"Artista Nashville","URL ":  
"https://www.artistanashville.com/labels"}  
{"id":5,"name ":"Elektra Records","URL ":  
"https://www.elektramusicgroup.com/"}
```

//REP REGION

```
{"id":1,"region_name ":"US-North"}  
{"id":2,"region_name ":"US-West"}  
{"id":3,"region_name ":"Canada"}  
{"id":4,"region_name ":"Asia"}  
{"id":5,"region_name ":"Europe"}  
{"id":6,"region_name ":"SouthAmerica"}
```

// CATEGORY

```
{"id":1,"desc ":"RB"}  
{"id":2,"desc ":"Jazz"}  
{"id":3,"desc ":"Metal Rock"}  
{"id":4,"desc ":"Alt Rock"}  
{"id":5,"desc ":"Easy Rock"}  
{"id":6,"desc ":"Classic Rock"}  
{"id":7,"desc ":"Hip-Hop"}
```