# Database Project – Final Report (Parts 1 to 5)

# Dixit Omkar

# Database Management Systems - Professor Fernandez

# George Washington University

# December 19, 2015

## Database Project Part 1

**Overview**

In this project objective was to design database system for catering company. Catering company has one hall and it has to manage orders. Also company has to keep track of orders and have different menus. Catering company also have contractors which can perform at ceremony. Customer has a choice whether to have ceremony in company hall or their own hall. Our database has menu starting from appetizers to desserts. It also has information about the location. The company also gives information about the florists that will be required and the musicians, entertainers and the photographers that will be required for the event. The company will give a complete itinerary that will be available to the client for calculating the cost. There are two tables that are combined. The cost table is dropped as the price of outside services is easily calculated from cost. Instead of using the client name we use the clientID as an ID for the event table.

## Purpose and objectives

The mission and vision consists on the following Objectives: To provide catering and event management services, that has not been experienced in the industry at affordable rates. To become the premier choice for corporate and government events, for event planning and brandling. To incorporate event management research and development functions in the company in the first five years. To achieve consistently a return on investment superior to all others to provide the basis for the long term capital growth. To be a company admired for its values and standards to all our clients. It takes up a lot of time to manually keep track of sales and place correct orders to vendors, wasting useful labor in trivial works. So we are developing a product which will help solving problems of the clients. This system can increase the efficiency of daily business transactions performed in the catering company by computerizing the entire current system. With this new computerized system many of the existing business processes such as total calculation, orders taking are done with the aid of computer. This will save time and error-free system will be created.

**Data Modeling Tools**

Oracle SQL Developer Data Modeler will be used to design the Data Flow Diagram, Entity Relationship Diagram, and Relational Model.

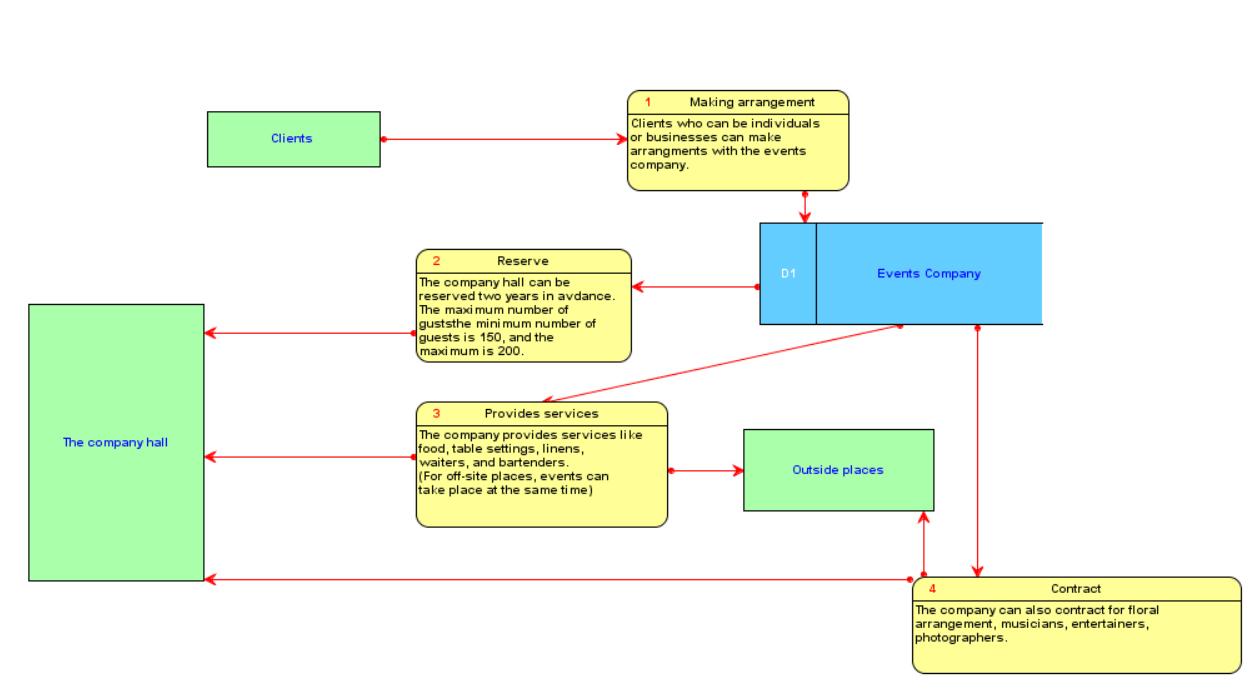
**Database Software**

The database will be based on Oracle 11g software.

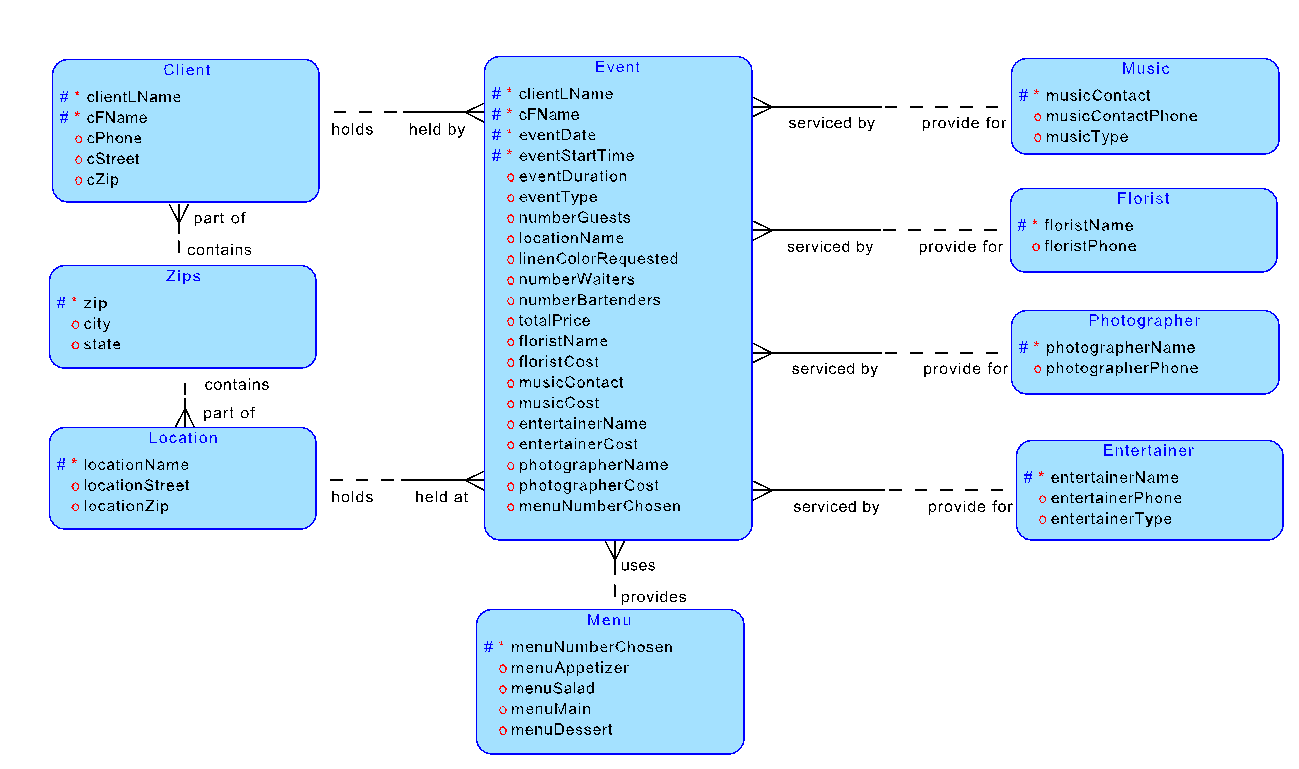
Oracle SQL Developer will be used to access the database although other Developer and Query Tools will also be compatible including SQL Plus, SQL Navigator and other SQL and PL/SQL developer tools.

## Database Project Part 2

**Data Flow Diagram (DFD)**



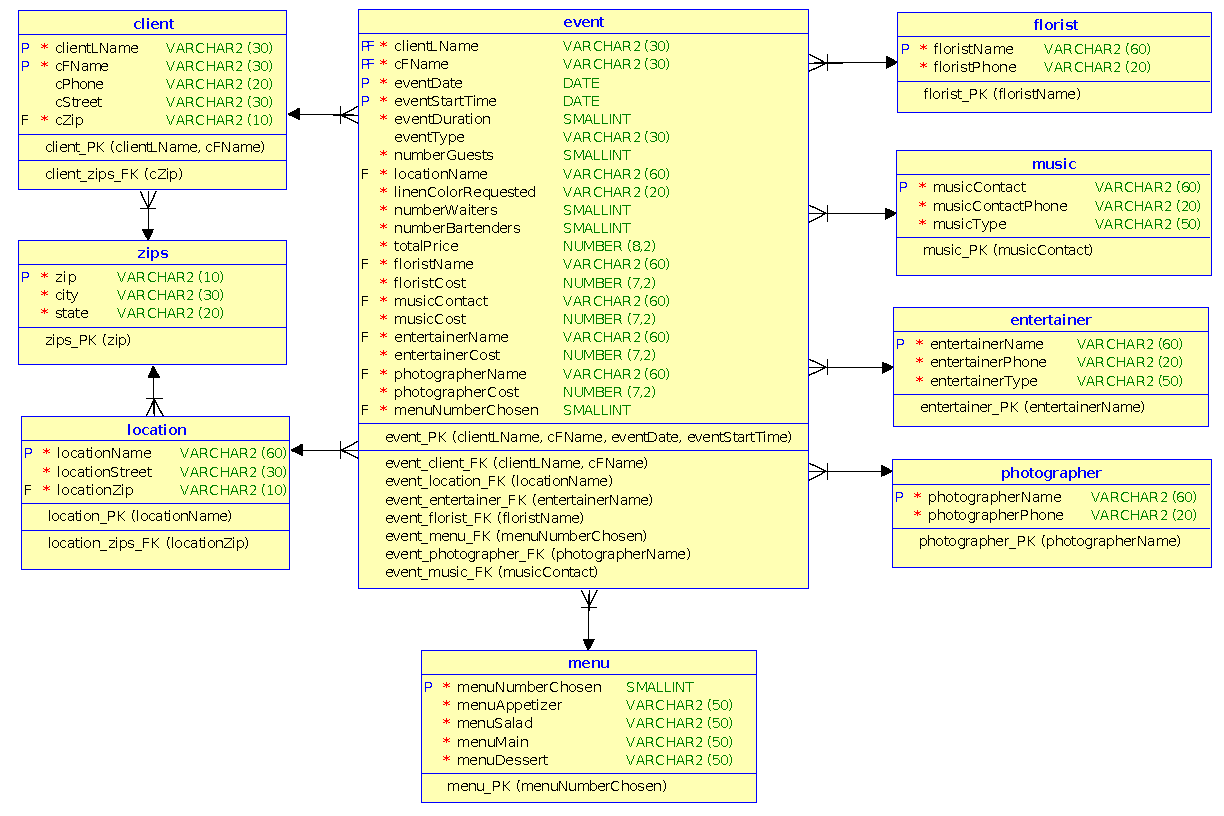
**Entity Relationship Diagram (ERD)**



­­­

## ­­Database Project Part 3

### a. Relational database design



### b. Build dependency diagrams

#### client table:

clientLName, cFName → cPhone, cStreet, cZip

#### zips table:

zip → city, state

#### location table:

locationName → locationStreet, locationZip

#### event table:

clientLName, cFName, eventDate, eventStartTime → eventDuration, eventType, numberGuests, locationName, linenColorRequested, numberWaiters, numberBartenders, totalPrice, floristName, floristCost, musicContact, musicCost, entertainerName, entertainerCost, photographerName, photographerCost, menuNumberChosen

#### menu table:

menuNumberChosen → menuAppetizer, menuSalad, menuMain, menuDessert

#### florist table:

floristName → floristPhone

#### music table:

musicContact → musicContactPhone, musicType

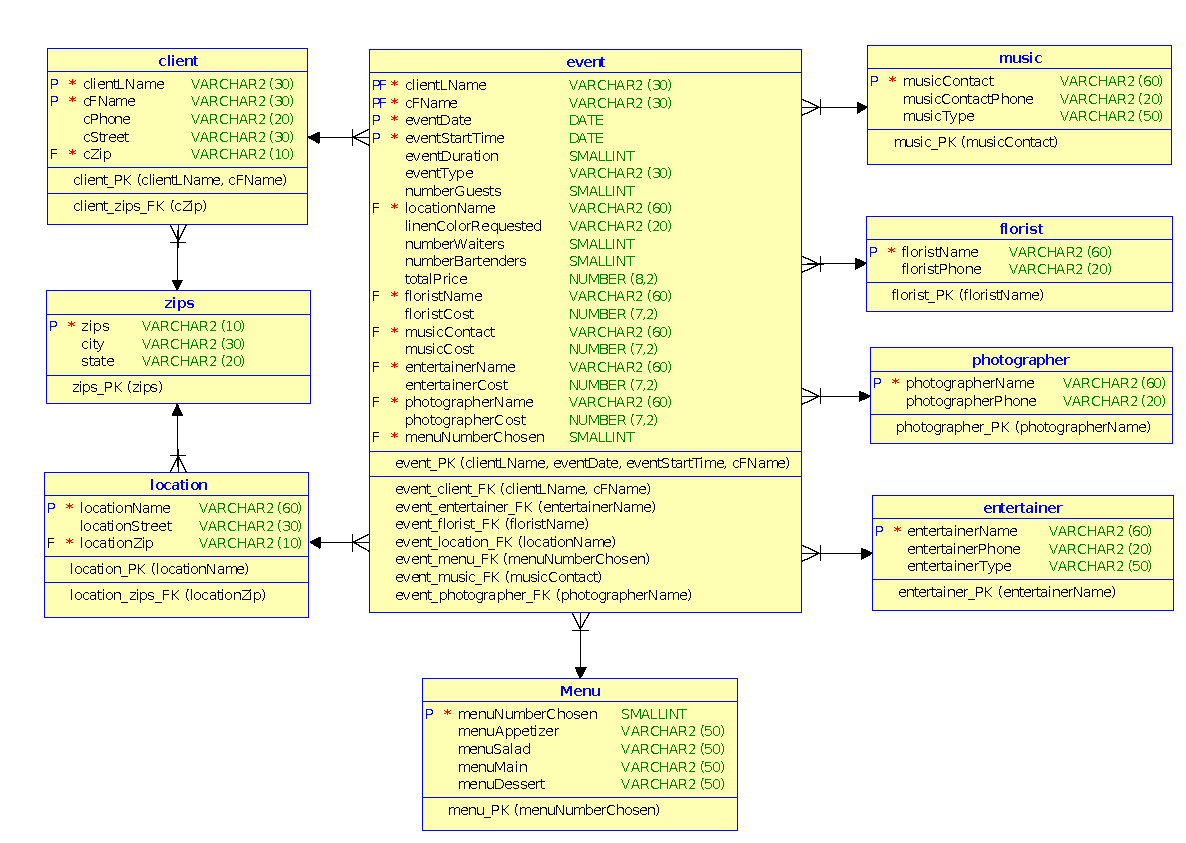
#### entertainer table:

entertainerName → entertainerPhone, entertainerType

#### photographer table:

photographerName → photographerPhone

### c. Use the Oracle data modeler to build a Relational Diagram



### d. Physical database design

|  |
| --- |
| -- Generated by Oracle SQL Developer Data Modeler 4.1.1.888  -- at: 2015-12-01 21:21:27 EST  -- site: Oracle Database 11g  -- type: Oracle Database 11g  CREATE TABLE Menu  (  menuNumberChosen SMALLINT NOT NULL ,  menuAppetizer VARCHAR2 (50) ,  menuSalad VARCHAR2 (50) ,  menuMain VARCHAR2 (50) ,  menuDessert VARCHAR2 (50)  ) ;  ALTER TABLE Menu ADD CONSTRAINT menu\_PK PRIMARY KEY ( menuNumberChosen ) ;  CREATE TABLE client  (  clientLName VARCHAR2 (30) CONSTRAINT NNC\_client\_clientLName NOT NULL ,  cFName VARCHAR2 (30) CONSTRAINT NNC\_client\_cFName NOT NULL ,  cPhone VARCHAR2 (20) ,  cStreet VARCHAR2 (30) ,  cZip VARCHAR2 (10) CONSTRAINT NNC\_client\_cZip NOT NULL  ) ;  ALTER TABLE client ADD CONSTRAINT client\_PK PRIMARY KEY ( clientLName, cFName ) ;  CREATE TABLE entertainer  (  entertainerName VARCHAR2 (60) NOT NULL ,  entertainerPhone VARCHAR2 (20) ,  entertainerType VARCHAR2 (50)  ) ;  ALTER TABLE entertainer ADD CONSTRAINT entertainer\_PK PRIMARY KEY ( entertainerName ) ;  CREATE TABLE event  (  clientLName VARCHAR2 (30) NOT NULL ,  cFName VARCHAR2 (30) NOT NULL ,  eventDate DATE NOT NULL ,  eventStartTime DATE NOT NULL ,  eventDuration SMALLINT ,  eventType VARCHAR2 (30) ,  numberGuests SMALLINT ,  locationName VARCHAR2 (60) NOT NULL ,  linenColorRequested VARCHAR2 (20) ,  numberWaiters SMALLINT ,  numberBartenders SMALLINT ,  totalPrice NUMBER (8,2) ,  floristName VARCHAR2 (60) NOT NULL ,  floristCost NUMBER (7,2) ,  musicContact VARCHAR2 (60) NOT NULL ,  musicCost NUMBER (7,2) ,  entertainerName VARCHAR2 (60) NOT NULL ,  entertainerCost NUMBER (7,2) ,  photographerName VARCHAR2 (60) NOT NULL ,  photographerCost NUMBER (7,2) ,  menuNumberChosen SMALLINT NOT NULL  ) ;  ALTER TABLE event ADD CONSTRAINT event\_PK PRIMARY KEY ( clientLName, eventDate, eventStartTime, cFName ) ;  CREATE TABLE florist  (  floristName VARCHAR2 (60) NOT NULL ,  floristPhone VARCHAR2 (20)  ) ;  ALTER TABLE florist ADD CONSTRAINT florist\_PK PRIMARY KEY ( floristName ) ;  CREATE TABLE location  (  locationName VARCHAR2 (60) NOT NULL ,  locationStreet VARCHAR2 (30) ,  locationZip VARCHAR2 (10) NOT NULL  ) ;  ALTER TABLE location ADD CONSTRAINT location\_PK PRIMARY KEY ( locationName ) ;  CREATE TABLE music  (  musicContact VARCHAR2 (60) NOT NULL ,  musicContactPhone VARCHAR2 (20) ,  musicType VARCHAR2 (50)  ) ;  ALTER TABLE music ADD CONSTRAINT music\_PK PRIMARY KEY ( musicContact ) ;  CREATE TABLE photographer  (  photographerName VARCHAR2 (60) NOT NULL ,  photographerPhone VARCHAR2 (20)  ) ;  ALTER TABLE photographer ADD CONSTRAINT photographer\_PK PRIMARY KEY ( photographerName ) ;  CREATE TABLE zips  (  zips VARCHAR2 (10) CONSTRAINT NNC\_zips\_zip NOT NULL ,  city VARCHAR2 (30) ,  state VARCHAR2 (20)  ) ;  ALTER TABLE zips ADD CONSTRAINT zips\_PK PRIMARY KEY ( zips ) ;  ALTER TABLE client ADD CONSTRAINT client\_zips\_FK FOREIGN KEY ( cZip ) REFERENCES zips ( zips ) ;  ALTER TABLE event ADD CONSTRAINT event\_client\_FK FOREIGN KEY ( clientLName, cFName ) REFERENCES client ( clientLName, cFName ) ;  ALTER TABLE event ADD CONSTRAINT event\_entertainer\_FK FOREIGN KEY ( entertainerName ) REFERENCES entertainer ( entertainerName ) ;  ALTER TABLE event ADD CONSTRAINT event\_florist\_FK FOREIGN KEY ( floristName ) REFERENCES florist ( floristName ) ;  ALTER TABLE event ADD CONSTRAINT event\_location\_FK FOREIGN KEY ( locationName ) REFERENCES location ( locationName ) ;  ALTER TABLE event ADD CONSTRAINT event\_menu\_FK FOREIGN KEY ( menuNumberChosen ) REFERENCES Menu ( menuNumberChosen ) ;  ALTER TABLE event ADD CONSTRAINT event\_music\_FK FOREIGN KEY ( musicContact ) REFERENCES music ( musicContact ) ;  ALTER TABLE event ADD CONSTRAINT event\_photographer\_FK FOREIGN KEY ( photographerName ) REFERENCES photographer ( photographerName ) ;  ALTER TABLE location ADD CONSTRAINT location\_zips\_FK FOREIGN KEY ( locationZip ) REFERENCES zips ( zips ) ; |

## Database Project Part 4:

For simpler design and to check whether hall is already occupied or not, I added attribute HALL\_TYPE.

If hall\_type=True then company’s hall will be used.

If hall\_type=False then clients own hall will be used.

cost\_hall field is added to the table event. This will be constant and it will be added to the total cost if hall\_type=1.

Misc\_cost attribute is added to calculate cost due to intangible factors such as how much busy they are on the date.

Menu\_cost attribute was added in menu as each menu item will have the corresponding cost.

Since all the events may not have florist, entertainer, new attribute services\_required attribute.

To provide facility of cancellation, order\_status attribute was added.

If order\_status is true then order is not cancelled and

If order\_status is false then order is not cancelled.

services\_required = False, None of the services are required.

services\_required = True, contractor company is required for florist, entertainer etc.

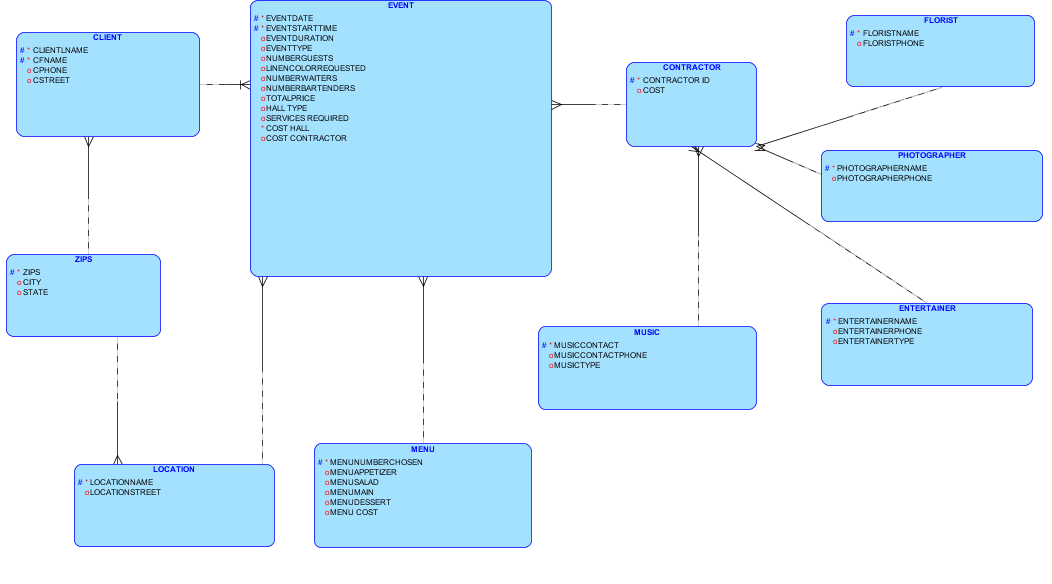
and contractor table was introduced in order to separate it from events table.

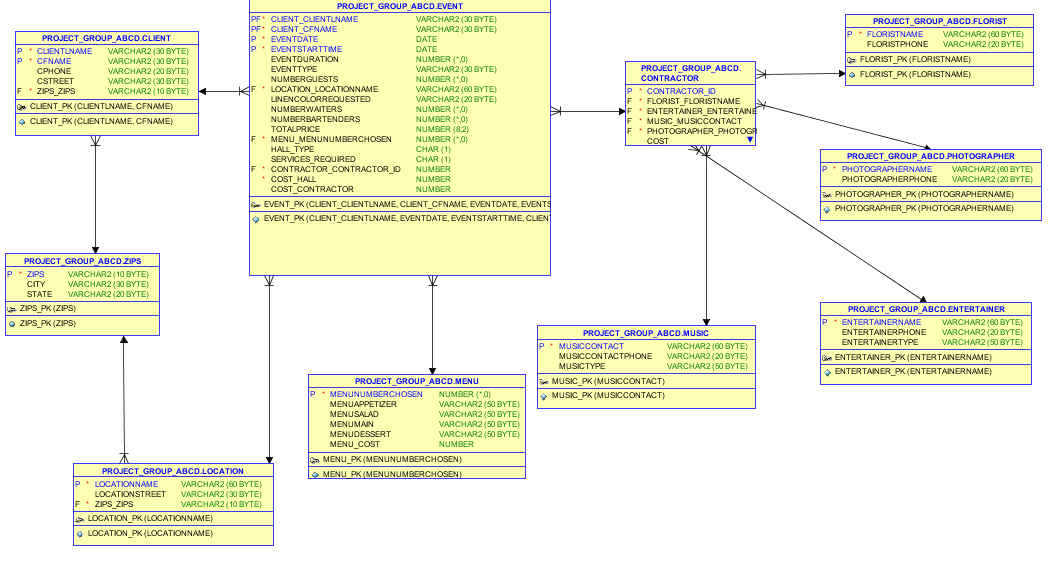
cost\_contractor field is added to the main events table event.

Totalcost will be calculated as –

**Modified Figures (based on changes described above)**

**Figure of Modified ERD**



**Figure of Modified Project Relational Diagram**

**Primary Keys, Foreign Keys, and Unique and Check Constraints:**

I added unique identifying attributes as primary keys. I also created foreign keys with identifying and linking fields of the two tables. I added following primary keys and foreign keys (to previous design)

CONTRCTOR\_ID is primary key for contractor.

FLORISTNAME is foreign key for contractor.

PHOTOGRAPHERNAME is foreign key for contractor.

MUSICIANNAME is foreign key for contractor.

ENTERTRAINERNAME is foreign key for contractor.

As mentioned in design there must be at least 150 guests and at max 200 guests, I added check constraint to the field NUMBERGUESTS field. But this constraint will be enforced if and only if hall of catering company is used that is, hall\_type = 1. As reservations can be done maximum 2 years in advance, check constraint was added to check whether reservation date is within 2 years of today’s date.

**Normalizations:**

Major normalization was done in 3rd phase itself. I tried to normalize more. The normalization process requires that all attributes are single valued (1NF), attribute needs to depend on that entity’s (table) unique identifier (2NF), and that no non-UID attribute can be dependent on another non-UID attribute (3NF) (Oracle, 2013b, 2013c, 2013d, and Coronel et al., 2011). There was only one change in the design and that was addition of table contractor. This was already on 3NF.

All the tables in the design are in 3NF.

**Sequences**

I created sequences in order to keep track of values which are not containing any useful business information. Following sequences were added into the design (see sql scripts)

SEQUENCE ORDERID\_SEQ ---------------------------to keep track of orders

SEQUENCE CONTRACTORID\_SEQ

SEQUNCE MENUNUMBERCHSEN\_SEQ

Value of each sequence is added by 1 when we add data in corresponding tables.

**Population of tables:**

I used insert statements to populate data. I added several orders in the table. I populated sequence using NEXTVAL. To avoid referential integrity constraints, I disabled the foreign key constraint and then inserted data and again I enabled foreign key constraint.

**What Did I learn in part 4:**

I learnt how to analyze requirement and add constraints in the tables accordingly. I also learned to normalize tables into 3rd normal form and Boycc-codd normal form. I learned to add auto incremental sequences. I also came to know about inserting data and how referential integrity needs to be maintained. I also learned about different types of joins and how to select data from different tables using outer, inner joins.

**DDL : (also present in zip file)**

-- Generated by Oracle SQL Developer Data Modeler 4.0.0.833

-- at: 2015-12-03 03:34:31 EST

-- site: Oracle Database 11g

-- type: Oracle Database 11g

CREATE USER PROJECT\_GROUP\_ABCD IDENTIFIED BY ACCOUNT UNLOCK ;

CREATE TABLE PROJECT\_GROUP\_ABCD.CLIENT

(

CLIENTLNAME VARCHAR2 (30 BYTE) CONSTRAINT NNC\_CLIENT\_CLIENTLNAME NOT NULL ,

CFNAME VARCHAR2 (30 BYTE) CONSTRAINT NNC\_CLIENT\_CFNAME NOT NULL ,

CPHONE VARCHAR2 (20 BYTE) ,

CSTREET VARCHAR2 (30 BYTE) ,

ZIPS\_ZIPS VARCHAR2 (10 BYTE) CONSTRAINT NNC\_CLIENT\_CZIP NOT NULL

)

PCTFREE 10 PCTUSED 40 TABLESPACE USERS LOGGING STORAGE

(

PCTINCREASE 0 MINEXTENTS 1 MAXEXTENTS UNLIMITED FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT

) ;

CREATE UNIQUE INDEX PROJECT\_GROUP\_ABCD.CLIENT\_PK ON PROJECT\_GROUP\_ABCD.CLIENT

(

CLIENTLNAME ASC , CFNAME ASC

)

TABLESPACE USERS PCTFREE 10 STORAGE

(

PCTINCREASE 0 MINEXTENTS 1 MAXEXTENTS UNLIMITED FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT

)

LOGGING ;

ALTER TABLE PROJECT\_GROUP\_ABCD.CLIENT ADD CONSTRAINT CLIENT\_PK PRIMARY KEY ( CLIENTLNAME, CFNAME ) USING INDEX PROJECT\_GROUP\_ABCD.CLIENT\_PK ;

CREATE TABLE PROJECT\_GROUP\_ABCD.CONTRACTOR

(

CONTRACTOR\_ID NUMBER NOT NULL ,

FLORIST\_FLORISTNAME VARCHAR2 (60 BYTE) NOT NULL ,

ENTERTAINER\_ENTERTAINERNAME VARCHAR2 (60 BYTE) NOT NULL ,

MUSIC\_MUSICCONTACT VARCHAR2 (60 BYTE) NOT NULL ,

PHOTOGRAPHER\_PHOTOGRAPHERNAME VARCHAR2 (60 BYTE) NOT NULL ,

COST NUMBER

)

LOGGING ;

ALTER TABLE PROJECT\_GROUP\_ABCD.CONTRACTOR ADD CONSTRAINT CONTRACTOR\_PK PRIMARY KEY ( CONTRACTOR\_ID ) ;

CREATE TABLE PROJECT\_GROUP\_ABCD.ENTERTAINER

(

ENTERTAINERNAME VARCHAR2 (60 BYTE) NOT NULL ,

ENTERTAINERPHONE VARCHAR2 (20 BYTE) ,

ENTERTAINERTYPE VARCHAR2 (50 BYTE)

)

PCTFREE 10 PCTUSED 40 TABLESPACE USERS LOGGING STORAGE

(

PCTINCREASE 0 MINEXTENTS 1 MAXEXTENTS UNLIMITED FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT

) ;

CREATE UNIQUE INDEX PROJECT\_GROUP\_ABCD.ENTERTAINER\_PK ON PROJECT\_GROUP\_ABCD.ENTERTAINER

(

ENTERTAINERNAME ASC

)

TABLESPACE USERS PCTFREE 10 STORAGE

(

PCTINCREASE 0 MINEXTENTS 1 MAXEXTENTS UNLIMITED FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT

)

LOGGING ;

ALTER TABLE PROJECT\_GROUP\_ABCD.ENTERTAINER ADD CONSTRAINT ENTERTAINER\_PK PRIMARY KEY ( ENTERTAINERNAME ) USING INDEX PROJECT\_GROUP\_ABCD.ENTERTAINER\_PK ;

CREATE TABLE PROJECT\_GROUP\_ABCD.EVENT

(

CLIENT\_CLIENTLNAME VARCHAR2 (30 BYTE) NOT NULL ,

CLIENT\_CFNAME VARCHAR2 (30 BYTE) NOT NULL ,

EVENTDATE DATE NOT NULL ,

EVENTSTARTTIME DATE NOT NULL ,

EVENTDURATION NUMBER (\*,0) ,

EVENTTYPE VARCHAR2 (30 BYTE) ,

NUMBERGUESTS NUMBER (\*,0) ,

LOCATION\_LOCATIONNAME VARCHAR2 (60 BYTE) NOT NULL ,

LINENCOLORREQUESTED VARCHAR2 (20 BYTE) ,

NUMBERWAITERS NUMBER (\*,0) ,

NUMBERBARTENDERS NUMBER (\*,0) ,

TOTALPRICE NUMBER (8,2) ,

MENU\_MENUNUMBERCHOSEN NUMBER (\*,0) NOT NULL ,

HALL\_TYPE CHAR (1) ,

SERVICES\_REQUIRED CHAR (1) ,

CONTRACTOR\_CONTRACTOR\_ID NUMBER NOT NULL ,

COST\_HALL NUMBER NOT NULL ,

COST\_CONTRACTOR NUMBER ,

ORDER\_ID NUMBER ,

ORDER\_STATUS CHAR (1)

)

PCTFREE 10 PCTUSED 40 TABLESPACE USERS LOGGING STORAGE

(

PCTINCREASE 0 MINEXTENTS 1 MAXEXTENTS UNLIMITED FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT

) ;

ALTER TABLE PROJECT\_GROUP\_ABCD.EVENT ADD CONSTRAINT NUMBEROFGUESTS\_CK CHECK (NUMBERGUESTS < 200 & NUMBERGUESTS > 150) ;

ALTER TABLE PROJECT\_GROUP\_ABCD.EVENT ADD CONSTRAINT EVENTDATE\_CK CHECK (eventdate < (sysdate + 730)) ;

CREATE UNIQUE INDEX PROJECT\_GROUP\_ABCD.EVENT\_PK ON PROJECT\_GROUP\_ABCD.EVENT ( CLIENT\_CLIENTLNAME ASC , EVENTDATE ASC , EVENTSTARTTIME ASC , CLIENT\_CFNAME ASC ) TABLESPACE USERS PCTFREE 10 STORAGE ( PCTINCREASE 0 MINEXTENTS 1 MAXEXTENTS UNLIMITED FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT ) LOGGING ;

ALTER TABLE PROJECT\_GROUP\_ABCD.EVENT ADD CONSTRAINT EVENT\_PK PRIMARY KEY ( CLIENT\_CLIENTLNAME, CLIENT\_CFNAME, EVENTDATE, EVENTSTARTTIME ) USING INDEX PROJECT\_GROUP\_ABCD.EVENT\_PK ;

CREATE TABLE PROJECT\_GROUP\_ABCD.FLORIST

(

FLORISTNAME VARCHAR2 (60 BYTE) NOT NULL ,

FLORISTPHONE VARCHAR2 (20 BYTE)

)

PCTFREE 10 PCTUSED 40 TABLESPACE USERS LOGGING STORAGE

(

PCTINCREASE 0 MINEXTENTS 1 MAXEXTENTS UNLIMITED FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT

) ;

CREATE UNIQUE INDEX PROJECT\_GROUP\_ABCD.FLORIST\_PK ON PROJECT\_GROUP\_ABCD.FLORIST

(

FLORISTNAME ASC

)

TABLESPACE USERS PCTFREE 10 STORAGE

(

PCTINCREASE 0 MINEXTENTS 1 MAXEXTENTS UNLIMITED FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT

)

LOGGING ;

ALTER TABLE PROJECT\_GROUP\_ABCD.FLORIST ADD CONSTRAINT FLORIST\_PK PRIMARY KEY ( FLORISTNAME ) USING INDEX PROJECT\_GROUP\_ABCD.FLORIST\_PK ;

CREATE TABLE PROJECT\_GROUP\_ABCD.LOCATION

(

LOCATIONNAME VARCHAR2 (60 BYTE) NOT NULL ,

LOCATIONSTREET VARCHAR2 (30 BYTE) ,

ZIPS\_ZIPS VARCHAR2 (10 BYTE) NOT NULL

)

PCTFREE 10 PCTUSED 40 TABLESPACE USERS LOGGING STORAGE

(

PCTINCREASE 0 MINEXTENTS 1 MAXEXTENTS UNLIMITED FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT

) ;

CREATE UNIQUE INDEX PROJECT\_GROUP\_ABCD.LOCATION\_PK ON PROJECT\_GROUP\_ABCD.LOCATION

(

LOCATIONNAME ASC

)

TABLESPACE USERS PCTFREE 10 STORAGE

(

PCTINCREASE 0 MINEXTENTS 1 MAXEXTENTS UNLIMITED FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT

)

LOGGING ;

ALTER TABLE PROJECT\_GROUP\_ABCD.LOCATION ADD CONSTRAINT LOCATION\_PK PRIMARY KEY ( LOCATIONNAME ) USING INDEX PROJECT\_GROUP\_ABCD.LOCATION\_PK ;

CREATE TABLE PROJECT\_GROUP\_ABCD.MENU

(

MENUNUMBERCHOSEN NUMBER (\*,0) NOT NULL ,

MENUAPPETIZER VARCHAR2 (50 BYTE) ,

MENUSALAD VARCHAR2 (50 BYTE) ,

MENUMAIN VARCHAR2 (50 BYTE) ,

MENUDESSERT VARCHAR2 (50 BYTE) ,

MENU\_COST NUMBER

)

PCTFREE 10 PCTUSED 40 TABLESPACE USERS LOGGING STORAGE

(

PCTINCREASE 0 MINEXTENTS 1 MAXEXTENTS UNLIMITED FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT

) ;

CREATE UNIQUE INDEX PROJECT\_GROUP\_ABCD.MENU\_PK ON PROJECT\_GROUP\_ABCD.MENU

(

MENUNUMBERCHOSEN ASC

)

TABLESPACE USERS PCTFREE 10 STORAGE

(

PCTINCREASE 0 MINEXTENTS 1 MAXEXTENTS UNLIMITED FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT

)

LOGGING ;

ALTER TABLE PROJECT\_GROUP\_ABCD.MENU ADD CONSTRAINT MENU\_PK PRIMARY KEY ( MENUNUMBERCHOSEN ) USING INDEX PROJECT\_GROUP\_ABCD.MENU\_PK ;

CREATE TABLE PROJECT\_GROUP\_ABCD.MUSIC

(

MUSICCONTACT VARCHAR2 (60 BYTE) NOT NULL ,

MUSICCONTACTPHONE VARCHAR2 (20 BYTE) ,

MUSICTYPE VARCHAR2 (50 BYTE)

)

PCTFREE 10 PCTUSED 40 TABLESPACE USERS LOGGING STORAGE

(

PCTINCREASE 0 MINEXTENTS 1 MAXEXTENTS UNLIMITED FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT

) ;

CREATE UNIQUE INDEX PROJECT\_GROUP\_ABCD.MUSIC\_PK ON PROJECT\_GROUP\_ABCD.MUSIC

(

MUSICCONTACT ASC

)

TABLESPACE USERS PCTFREE 10 STORAGE

(

PCTINCREASE 0 MINEXTENTS 1 MAXEXTENTS UNLIMITED FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT

)

LOGGING ;

ALTER TABLE PROJECT\_GROUP\_ABCD.MUSIC ADD CONSTRAINT MUSIC\_PK PRIMARY KEY ( MUSICCONTACT ) USING INDEX PROJECT\_GROUP\_ABCD.MUSIC\_PK ;

CREATE TABLE PROJECT\_GROUP\_ABCD.PHOTOGRAPHER

(

PHOTOGRAPHERNAME VARCHAR2 (60 BYTE) NOT NULL ,

PHOTOGRAPHERPHONE VARCHAR2 (20 BYTE)

)

PCTFREE 10 PCTUSED 40 TABLESPACE USERS LOGGING STORAGE

(

PCTINCREASE 0 MINEXTENTS 1 MAXEXTENTS UNLIMITED FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT

) ;

CREATE UNIQUE INDEX PROJECT\_GROUP\_ABCD.PHOTOGRAPHER\_PK ON PROJECT\_GROUP\_ABCD.PHOTOGRAPHER

(

PHOTOGRAPHERNAME ASC

)

TABLESPACE USERS PCTFREE 10 STORAGE

(

PCTINCREASE 0 MINEXTENTS 1 MAXEXTENTS UNLIMITED FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT

)

LOGGING ;

ALTER TABLE PROJECT\_GROUP\_ABCD.PHOTOGRAPHER ADD CONSTRAINT PHOTOGRAPHER\_PK PRIMARY KEY ( PHOTOGRAPHERNAME ) USING INDEX PROJECT\_GROUP\_ABCD.PHOTOGRAPHER\_PK ;

CREATE TABLE PROJECT\_GROUP\_ABCD.ZIPS

(

ZIPS VARCHAR2 (10 BYTE) CONSTRAINT NNC\_ZIPS\_ZIP NOT NULL ,

CITY VARCHAR2 (30 BYTE) ,

STATE VARCHAR2 (20 BYTE)

)

PCTFREE 10 PCTUSED 40 TABLESPACE USERS LOGGING STORAGE

(

PCTINCREASE 0 MINEXTENTS 1 MAXEXTENTS UNLIMITED FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT

) ;

CREATE UNIQUE INDEX PROJECT\_GROUP\_ABCD.ZIPS\_PK ON PROJECT\_GROUP\_ABCD.ZIPS

(

ZIPS ASC

)

TABLESPACE USERS PCTFREE 10 STORAGE

(

PCTINCREASE 0 MINEXTENTS 1 MAXEXTENTS UNLIMITED FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT

)

LOGGING ;

ALTER TABLE PROJECT\_GROUP\_ABCD.ZIPS ADD CONSTRAINT ZIPS\_PK PRIMARY KEY ( ZIPS ) USING INDEX PROJECT\_GROUP\_ABCD.ZIPS\_PK ;

ALTER TABLE PROJECT\_GROUP\_ABCD.CLIENT ADD CONSTRAINT CLIENT\_ZIPS\_FK FOREIGN KEY ( ZIPS\_ZIPS ) REFERENCES PROJECT\_GROUP\_ABCD.ZIPS ( ZIPS ) NOT DEFERRABLE ;

ALTER TABLE PROJECT\_GROUP\_ABCD.CONTRACTOR ADD CONSTRAINT CONTRACTOR\_ENTERTAINER\_FK FOREIGN KEY ( ENTERTAINER\_ENTERTAINERNAME ) REFERENCES PROJECT\_GROUP\_ABCD.ENTERTAINER ( ENTERTAINERNAME ) NOT DEFERRABLE ;

ALTER TABLE PROJECT\_GROUP\_ABCD.CONTRACTOR ADD CONSTRAINT CONTRACTOR\_FLORIST\_FK FOREIGN KEY ( FLORIST\_FLORISTNAME ) REFERENCES PROJECT\_GROUP\_ABCD.FLORIST ( FLORISTNAME ) NOT DEFERRABLE ;

ALTER TABLE PROJECT\_GROUP\_ABCD.CONTRACTOR ADD CONSTRAINT CONTRACTOR\_MUSIC\_FK FOREIGN KEY ( MUSIC\_MUSICCONTACT ) REFERENCES PROJECT\_GROUP\_ABCD.MUSIC ( MUSICCONTACT ) NOT DEFERRABLE ;

ALTER TABLE PROJECT\_GROUP\_ABCD.CONTRACTOR ADD CONSTRAINT CONTRACTOR\_PHOTOGRAPHER\_FK FOREIGN KEY ( PHOTOGRAPHER\_PHOTOGRAPHERNAME ) REFERENCES PROJECT\_GROUP\_ABCD.PHOTOGRAPHER ( PHOTOGRAPHERNAME ) NOT DEFERRABLE ;

ALTER TABLE PROJECT\_GROUP\_ABCD.EVENT ADD CONSTRAINT EVENT\_CLIENT\_FK FOREIGN KEY ( CLIENT\_CLIENTLNAME, CLIENT\_CFNAME ) REFERENCES PROJECT\_GROUP\_ABCD.CLIENT ( CLIENTLNAME, CFNAME ) NOT DEFERRABLE ;

ALTER TABLE PROJECT\_GROUP\_ABCD.EVENT ADD CONSTRAINT EVENT\_CONTRACTOR\_FK FOREIGN KEY ( CONTRACTOR\_CONTRACTOR\_ID ) REFERENCES PROJECT\_GROUP\_ABCD.CONTRACTOR ( CONTRACTOR\_ID ) NOT DEFERRABLE ;

ALTER TABLE PROJECT\_GROUP\_ABCD.EVENT ADD CONSTRAINT EVENT\_LOCATION\_FK FOREIGN KEY ( LOCATION\_LOCATIONNAME ) REFERENCES PROJECT\_GROUP\_ABCD.LOCATION ( LOCATIONNAME ) NOT DEFERRABLE ;

ALTER TABLE PROJECT\_GROUP\_ABCD.EVENT ADD CONSTRAINT EVENT\_MENU\_FK FOREIGN KEY ( MENU\_MENUNUMBERCHOSEN ) REFERENCES PROJECT\_GROUP\_ABCD.MENU ( MENUNUMBERCHOSEN ) NOT DEFERRABLE ;

ALTER TABLE PROJECT\_GROUP\_ABCD.LOCATION ADD CONSTRAINT LOCATION\_ZIPS\_FK FOREIGN KEY ( ZIPS\_ZIPS ) REFERENCES PROJECT\_GROUP\_ABCD.ZIPS ( ZIPS ) NOT DEFERRABLE ;

CREATE SEQUENCE PROJECT\_GROUP\_ABCD.CONTRACTOR\_CONTRACTOR\_ID\_SEQ START WITH 1 NOCACHE ORDER ;

CREATE OR REPLACE TRIGGER PROJECT\_GROUP\_ABCD.CONTRACTOR\_CONTRACTOR\_ID\_TRG BEFORE

INSERT ON PROJECT\_GROUP\_ABCD.CONTRACTOR FOR EACH ROW WHEN (NEW.CONTRACTOR\_ID IS NULL) BEGIN :NEW.CONTRACTOR\_ID := PROJECT\_GROUP\_ABCD.CONTRACTOR\_CONTRACTOR\_ID\_SEQ.NEXTVAL;

END;

/

CREATE SEQUENCE PROJECT\_GROUP\_ABCD.EVENT\_ORDER\_ID\_SEQ START WITH 1 NOCACHE ORDER ;

CREATE OR REPLACE TRIGGER PROJECT\_GROUP\_ABCD.EVENT\_ORDER\_ID\_TRG BEFORE

INSERT ON PROJECT\_GROUP\_ABCD.EVENT FOR EACH ROW WHEN (NEW.ORDER\_ID IS NULL) BEGIN :NEW.ORDER\_ID := PROJECT\_GROUP\_ABCD.EVENT\_ORDER\_ID\_SEQ.NEXTVAL;

END;

/

CREATE SEQUENCE PROJECT\_GROUP\_ABCD.MENU\_MENUNUMBERCHOSEN\_SEQ START WITH 1 NOCACHE ORDER ;

CREATE OR REPLACE TRIGGER PROJECT\_GROUP\_ABCD.MENU\_MENUNUMBERCHOSEN\_TRG BEFORE

INSERT ON PROJECT\_GROUP\_ABCD.MENU FOR EACH ROW WHEN (NEW.MENUNUMBERCHOSEN IS NULL) BEGIN :NEW.MENUNUMBERCHOSEN := PROJECT\_GROUP\_ABCD.MENU\_MENUNUMBERCHOSEN\_SEQ.NEXTVAL;

END;

/

-- Oracle SQL Developer Data Modeler Summary Report:

--

-- CREATE TABLE 10

-- CREATE INDEX 9

-- ALTER TABLE 22

-- CREATE VIEW 0

-- CREATE PACKAGE 0

-- CREATE PACKAGE BODY 0

-- CREATE PROCEDURE 0

-- CREATE FUNCTION 0

-- CREATE TRIGGER 3

-- ALTER TRIGGER 0

-- CREATE COLLECTION TYPE 0

-- CREATE STRUCTURED TYPE 0

-- CREATE STRUCTURED TYPE BODY 0

-- CREATE CLUSTER 0

-- CREATE CONTEXT 0

-- CREATE DATABASE 0

-- CREATE DIMENSION 0

-- CREATE DIRECTORY 0

-- CREATE DISK GROUP 0

-- CREATE ROLE 0

-- CREATE ROLLBACK SEGMENT 0

-- CREATE SEQUENCE 3

-- CREATE MATERIALIZED VIEW 0

-- CREATE SYNONYM 0

-- CREATE TABLESPACE 0

-- CREATE USER 1

--

-- DROP TABLESPACE 0

-- DROP DATABASE 0

--

-- REDACTION POLICY 0

--

-- ERRORS 0

-- WARNINGS 0

***Location***

Insert into LOCATION (LOCATIONNAME,LOCATIONSTREET,ZIPS\_ZIPS) values ('Columbia Plaza','Virginai Avenue NW','20037');

Insert into LOCATION (LOCATIONNAME,LOCATIONSTREET,ZIPS\_ZIPS) values ('Louden county','Woods road','20101');

Insert into LOCATION (LOCATIONNAME,LOCATIONSTREET,ZIPS\_ZIPS) values ('Satnderd','Rappahannock','22747');

Insert into LOCATION (LOCATIONNAME,LOCATIONSTREET,ZIPS\_ZIPS) values ('Beach','Virginia Beach','23450');

Insert into LOCATION (LOCATIONNAME,LOCATIONSTREET,ZIPS\_ZIPS) values ('City','Virginia Beach city','23479');

Insert into LOCATION (LOCATIONNAME,LOCATIONSTREET,ZIPS\_ZIPS) values ('Florist','Florist st','20022');

Insert into LOCATION (LOCATIONNAME,LOCATIONSTREET,ZIPS\_ZIPS) values ('Beach2','Virginia Bch','23455');

Insert into LOCATION (LOCATIONNAME,LOCATIONSTREET,ZIPS\_ZIPS) values ('Louden county2','Ryan road','20107');

***Zips***

Insert into ZIPS (ZIPS,CITY,STATE) values ('20022','Washington','District of Columbia');

Insert into ZIPS (ZIPS,CITY,STATE) values ('20101','Dulles','District of Columbia');

Insert into ZIPS (ZIPS,CITY,STATE) values ('20107','Arcola','Loudoun County');

Insert into ZIPS (ZIPS,CITY,STATE) values ('22747','Washington','District of Columbia');

Insert into ZIPS (ZIPS,CITY,STATE) values ('23450','Virgina','District of Columbia');

Insert into ZIPS (ZIPS,CITY,STATE) values ('23455','Virginia','District of Columbia');

Insert into ZIPS (ZIPS,CITY,STATE) values ('23479','Virginia','District of Columbia');

Insert into ZIPS (ZIPS,CITY,STATE) values ('20037','Washington','District of Columbia');

***Client***

Insert into CLIENT (CLIENTLNAME,CFNAME,CPHONE,CSTREET,ZIPS\_ZIPS) values ('Omkar','dominos','20245621000',null,'20037');

Insert into CLIENT (CLIENTLNAME,CFNAME,CPHONE,CSTREET,ZIPS\_ZIPS) values ('Aditya','macdonalds','20245621002',null,'20022');

Insert into CLIENT (CLIENTLNAME,CFNAME,CPHONE,CSTREET,ZIPS\_ZIPS) values ('Ishan','subway','20245621003',null,'22747');

Insert into CLIENT (CLIENTLNAME,CFNAME,CPHONE,CSTREET,ZIPS\_ZIPS) values ('Madhura','chipotale','20245621400',null,'23450');

Insert into CLIENT (CLIENTLNAME,CFNAME,CPHONE,CSTREET,ZIPS\_ZIPS) values ('Malik','phalafal','20245621050',null,'23455');

Insert into CLIENT (CLIENTLNAME,CFNAME,CPHONE,CSTREET,ZIPS\_ZIPS) values ('David','pizza hut','20245621600',null,'23479');

Insert into CLIENT (CLIENTLNAME,CFNAME,CPHONE,CSTREET,ZIPS\_ZIPS) values ('Vishnu','donuts','20245621070',null,'20101');

Insert into CLIENT (CLIENTLNAME,CFNAME,CPHONE,CSTREET,ZIPS\_ZIPS) values ('Afreen','cheescake','20245621800',null,'20107');

***Menu***

insert into menu values(MENU\_MENUNUMBERCHOSEN\_SEQ.nextval,'cheese bread','caesar salad','chipotle','Luscious lemon',20);

insert into menu values(MENU\_MENUNUMBERCHOSEN\_SEQ.nextval,'raodside slider','laua salad','pizza','cookie tuffle',22);

insert into menu values(MENU\_MENUNUMBERCHOSEN\_SEQ.nextval,'chicken pot stickers','avocado salad','chicken','brownie',30);

insert into menu values(MENU\_MENUNUMBERCHOSEN\_SEQ.nextval,'avocado eggrolls','caesar salad','paneer','fruit pizza',10);

insert into menu values(MENU\_MENUNUMBERCHOSEN\_SEQ.nextval,'raodside slide','rlaua salad','pizza','tuffle',15);

insert into menu values(MENU\_MENUNUMBERCHOSEN\_SEQ.nextval,'cheese bread','avocado salad','sandwitch','brownie',8);

***Contractor***

alter table contractor disable constraint CONTRACTOR\_ENTERTAINER\_FK;

alter table contractor disable constraint CONTRACTOR\_FLORIST\_FK;

alter table contractor disable constraint CONTRACTOR\_MUSIC\_FK;

alter table contractor disable constraint CONTRACTOR\_PHOTOGRAPHER\_FK;

insert into contractor values(CONTRACTOR\_CONTRACTOR\_ID\_SEQ.nextval,'Sachin','Ronaldo','Ronaldo','Vishwanathan',150);

insert into contractor values(CONTRACTOR\_CONTRACTOR\_ID\_SEQ.nextval,'Ricky','Messi','Roger','Gary',150);

alter table contractor enable constraint CONTRACTOR\_ENTERTAINER\_FK;

alter table contractor enable constraint CONTRACTOR\_FLORIST\_FK;

alter table contractor enable constraint CONTRACTOR\_MUSIC\_FK;

alter table contractor enable constraint CONTRACTOR\_PHOTOGRAPHER\_FK;

Florist

insert into florist values('Sachin','2024501234');

insert into florist values('Ricky','2024511234');

Music

insert into music values('Sachin','2034501234','Rock');

insert into music values('Ricky','2025511234','Classical');

***Photographer***

insert into photographer values('Gary','2034501234');

insert into photographer values('Vishwanathan','2024561234');

***Entertainer***

insert into ENTERTAINER values('Ronaldo','2434501234','Dance');

insert into ENTERTAINER values('Messi','2225511234','Standup comedy');

***Event***

Insert into EVENT (CLIENT\_CLIENTLNAME,CLIENT\_CFNAME,EVENTDATE,EVENTSTARTTIME,EVENTDURATION,EVENTTYPE,NUMBERGUESTS,LOCATION\_LOCATIONNAME,LINENCOLORREQUESTED,NUMBERWAITERS,NUMBERBARTENDERS,TOTALPRICE,MENU\_MENUNUMBERCHOSEN,HALL\_TYPE,SERVICES\_REQUIRED,COST\_HALL,CONTRACTOR\_CONTRACTOR\_ID,COST\_CONTRACTOR,ORDER\_ID,ORDER\_STATUS) values ('Omkar','dominos',to\_date('25-12-15','DD-MM-RR'),to\_date('25-12-15','DD-MM-RR'),2,'Entertainment',160,'Columbia Plaza','Red',10,10,2000,1,'0','0',100,10,0,1,'T');

Insert into EVENT (CLIENT\_CLIENTLNAME,CLIENT\_CFNAME,EVENTDATE,EVENTSTARTTIME,EVENTDURATION,EVENTTYPE,NUMBERGUESTS,LOCATION\_LOCATIONNAME,LINENCOLORREQUESTED,NUMBERWAITERS,NUMBERBARTENDERS,TOTALPRICE,MENU\_MENUNUMBERCHOSEN,HALL\_TYPE,SERVICES\_REQUIRED,COST\_HALL,CONTRACTOR\_CONTRACTOR\_ID,COST\_CONTRACTOR,ORDER\_ID,ORDER\_STATUS) values ('Aditya','macdonalds',to\_date('26-12-15','DD-MM-RR'),to\_date('26-12-15','DD-MM-RR'),2,'Entertainment',170,'Columbia Plaza','Blue',12,10,2200,2,'0','0',100,11,0,2,'T');

Insert into EVENT (CLIENT\_CLIENTLNAME,CLIENT\_CFNAME,EVENTDATE,EVENTSTARTTIME,EVENTDURATION,EVENTTYPE,NUMBERGUESTS,LOCATION\_LOCATIONNAME,LINENCOLORREQUESTED,NUMBERWAITERS,NUMBERBARTENDERS,TOTALPRICE,MENU\_MENUNUMBERCHOSEN,HALL\_TYPE,SERVICES\_REQUIRED,COST\_HALL,CONTRACTOR\_CONTRACTOR\_ID,COST\_CONTRACTOR,ORDER\_ID,ORDER\_STATUS) values ('Ishan','subway',to\_date('27-12-15','DD-MM-RR'),to\_date('27-12-15','DD-MM-RR'),3,'Business',180,'Satnderd','Yellow',11,2,1500,3,'0','0',100,10,0,3,'T');

Insert into EVENT (CLIENT\_CLIENTLNAME,CLIENT\_CFNAME,EVENTDATE,EVENTSTARTTIME,EVENTDURATION,EVENTTYPE,NUMBERGUESTS,LOCATION\_LOCATIONNAME,LINENCOLORREQUESTED,NUMBERWAITERS,NUMBERBARTENDERS,TOTALPRICE,MENU\_MENUNUMBERCHOSEN,HALL\_TYPE,SERVICES\_REQUIRED,COST\_HALL,CONTRACTOR\_CONTRACTOR\_ID,COST\_CONTRACTOR,ORDER\_ID,ORDER\_STATUS) values ('Madhura','chipotale',to\_date('28-12-15','DD-MM-RR'),to\_date('28-12-15','DD-MM-RR'),1,'Business',190,'Beach','Blue',8,9,1200,4,'0','0',100,10,0,4,'T');

Insert into EVENT (CLIENT\_CLIENTLNAME,CLIENT\_CFNAME,EVENTDATE,EVENTSTARTTIME,EVENTDURATION,EVENTTYPE,NUMBERGUESTS,LOCATION\_LOCATIONNAME,LINENCOLORREQUESTED,NUMBERWAITERS,NUMBERBARTENDERS,TOTALPRICE,MENU\_MENUNUMBERCHOSEN,HALL\_TYPE,SERVICES\_REQUIRED,COST\_HALL,CONTRACTOR\_CONTRACTOR\_ID,COST\_CONTRACTOR,ORDER\_ID,ORDER\_STATUS) values ('David','pizza hut',to\_date('28-12-15','DD-MM-RR'),to\_date('28-12-15','DD-MM-RR'),3,'Entertainment',155,'Beach','Blue',12,12,1800,1,'0','1',100,11,300,5,'T');

Insert into EVENT (CLIENT\_CLIENTLNAME,CLIENT\_CFNAME,EVENTDATE,EVENTSTARTTIME,EVENTDURATION,EVENTTYPE,NUMBERGUESTS,LOCATION\_LOCATIONNAME,LINENCOLORREQUESTED,NUMBERWAITERS,NUMBERBARTENDERS,TOTALPRICE,MENU\_MENUNUMBERCHOSEN,HALL\_TYPE,SERVICES\_REQUIRED,COST\_HALL,CONTRACTOR\_CONTRACTOR\_ID,COST\_CONTRACTOR,ORDER\_ID,ORDER\_STATUS) values ('Vishnu','donuts',to\_date('28-12-15','DD-MM-RR'),to\_date('28-12-15','DD-MM-RR'),2,'Business',155,'Louden county2','Blue',2,5,800,2,'0','1',100,10,300,6,'T');

**Select statements:**

*To find out number of events between 20dec to 27 dec.*

select count(\*) from EVENT where eventdate between '20122015' and '27122015';

3

*To find out how many contracts were given between 20dec to 28 dec.*

select count(\*) from EVENT where eventdate between '20122015' and '28122015' and services\_required=1;

2

*To find out total profit earned between 20dec to 28dec.*

select sum(totalprice) from EVENT where eventdate between '20122015' and '28122015'

9500

*To find out photographer name for order number 5.*

select c.PHOTOGRAPHER\_PHOTOGRAPHERNAME from event e join contractor c on (c.contractor\_id=e.contractor\_contractor\_id) where e.order\_id=4;

Vishwanathan

*To find out cost of menu which was chosen for order number 4.*

select p.menu\_cost from event e join menu p on (e.menu\_menunumberchosen= p.menunumberchosen) where e.ORDER\_ID=4;

10

*To find out client phone number for the event on 20151225.*

select c.cphone from event e join client c on (c.CLIENTLNAME=e.CLIENT\_CLIENTLNAME) where e.EVENTDATE='25122015';

20245621000

*To find out musician name performing for order number 2*

select c.MUSIC\_MUSICCONTACT from event e join contractor c on (c.contractor\_id=e.contractor\_contractor\_id) where e.order\_id=2;

Roger

*To find out number of guests arriving on 20151226.*

select numberguests from event where eventdate='26122015';

170

*To find out number of bartenders required on 20151225.*

select sum(numberbartenders) from event where eventdate='26122015';

10

*To find entertainer name performing for order number 3.*

select c.entertainer\_entertainername from event e join contractor c on (c.contractor\_id=e.contractor\_contractor\_id) where e.order\_id=3;

Ronaldo

## Database Project Part 5

**Trigger Creation and Evaluation**

Part 5 of the database project focused on the creation of database triggers. Triggers are used to perform actions when an event occurs, or is “triggered”, in the database management system (Oracle, 2013b, Coronel et al. 2011). For this part of the project, we were required to create several triggers and to evaluate their functionality using SQL queries to create and update data (Oracle, 2013c, 2013d), and to evaluate the contents of a history table used to store old records replaced during a recent update. This part of the project also included the creation of several audit variables used to store user and date information to track who and when data was created or updated (Oracle, 2013b, 2013c, 2013d).

Five triggers were created for the database. One trigger was created for the purpose of saving old records in CLIENTHIST\_TRIG. This trigger, CLIENTHIST\_TRIG, saves the old values of the row being updated into an CLIENT table. The CLIENT table is a very important component of this database and the table ensures that recently changed rows will be stored for later use if they are needed after committed changes are made to CLIENT.

Four additional triggers were created to record audit information for two different tables. This information is stored in two audit attributes that were added to four tables. The audit attributes, created\_by, created\_date, updated\_by, and updated\_date were added to the two tables CLIENT, ENTERTAINER. These new table attributes are used by the following two triggers:

CLIENT\_AUD\_TRIG

ENTERTAINERAUD\_TRIG

These two additional triggers and the audit attributes are used to automatically store audit information for the user creating rows or making changes to rows, and the dates of these row inserts or changes.