Assignment 9

Part 1: Scenario Description

A manager of a science and technology museum is building a database to record visitors' activities on each opening date. The data of visitors' basic personal information is recorded when they buy their tickets for certain entertainments. The entering time and date for certain entertainments of a visitor are also recorded automatically by electronic devices.

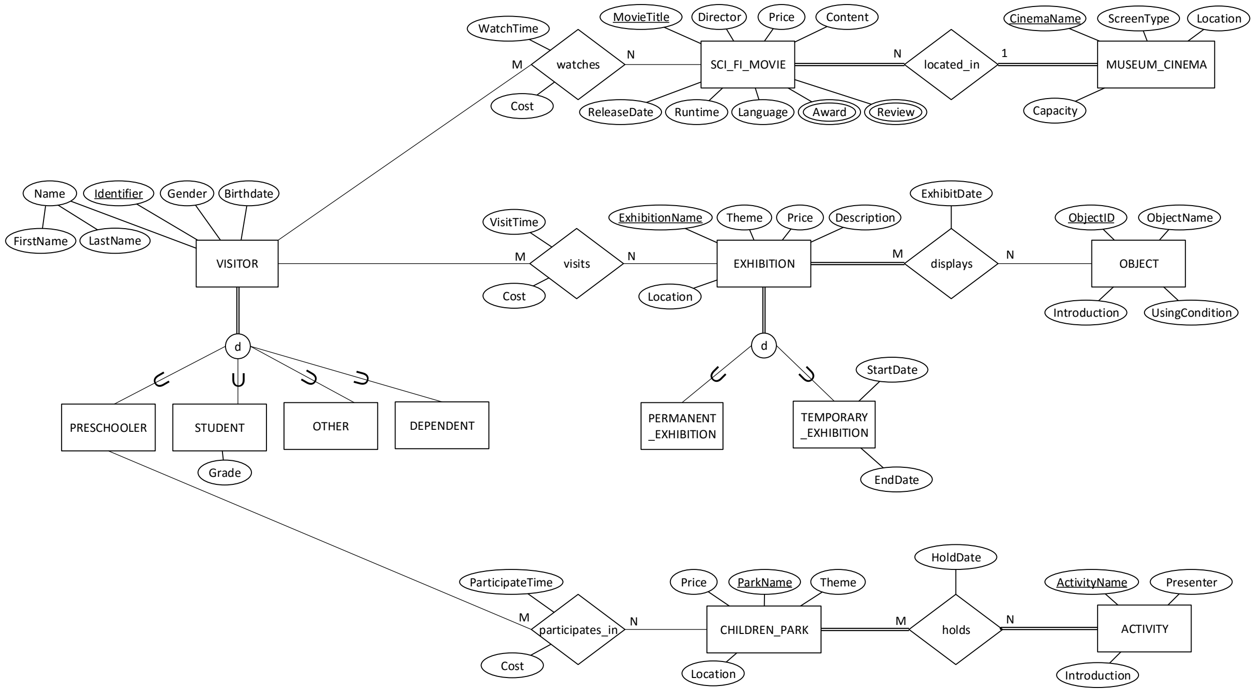
For each visitor, the database should record the Name (FirstName, LastName), Identifier (primary key), Gender and Birthdate. A visitor can have several roles: preschooler, student, other and dependent. Each visitor must have and only have one role. For each student, Grade should also be recorded. Each preschooler must be accompanied by at least one dependent, and a dependent can be responsible for more than one child.

Visitors can watch many sci-fi movies in different cinemas in the museum. The time they watched some movie and the cost they paid for it need to be recorded. Sci-fi movies have many attributes, including MovieTitle (primary key), Director, Price, Content, ReleaseDate, Runtime, Language, Award (multi-valued) and Review (multi-valued). Movies of the same screen type are played in one cinema, and one cinema can only play one type movies. In other words, a movie can only be located in one cinema. For example, 4D movies can only be played in the 4D cinema. The attributes of museum's cinemas are CinemaName (primary key), ScreenType, Location and Capacity.

Most visitors come to the museum for visiting kinds of exhibitions. Their VisitTime and Cost should be recorded in the database. The exhibitions have different ExhibitionName (primary key), Theme, Price, Description and Location. Exhibitions can be classified into two subclasses: permanent exhibitions and temporary exhibitions which have StartDate and EndDate. Objects for display can be set in multiple exhibitions on different dates. And some objects are under repairing, which means they are not being displayed. Each object has the following attributes: ObjectID (primary key), ObjectName, UsingCondition, and Introduction.

The subclass preschoolers can also participate in the children park. The participating time and the cost should be recorded. There are many kinds of children parks for selection. Each children park has ParkName (primary key), Price, Theme and Location. Besides, every children park can hold many kinds of activities on different dates. The attributes of activities are ActivityName (primary key), Presenter and Introduction.

Part 2: EER Diagram



Part 3: Relational Database Schema

Step 1: For Superclass/Subclass Entities

Superclass/Subclasses	Choices Available	Choice Made and Why
VISITOR / PRESCHOOLER, STUDENT, OTHER, DEPENDENT	8A, 8B, 8C, 8D (NOT RECOMMAND)	8C. Because there is only one local attribute and there is no complex relationship between these subclasses.
EXHIBITION / PERMANENT_EXHIBITION, TEMPORARY_EXHIBITION	8A, 8B, 8C, 8D (NOT RECOMMAND)	8C. Because there are only two local attributes of one subclass and the subclasses do not have their own respective relationship.

Step 2: Map out Strong Entities

In addition to the two superclass/subclass entities, there are 5 strong entities. They are SCI_FI_MOVIE, MUSEUM_CINEMA, OBJECT, CHILDREN_PARK and ACTIVITY. Create a new table for each of them.

Step 3: Map out Weak Entities

There is no weak entity.

Step 4: 1 to 1 Relationship

There aren’t any.

Step 5: 1 to N Relationship

Put the primary key of table MUSEUM_CINEMA as a foreign key of table SCI_FI_MOVIE.

Step 6: M to N Relationship

There are 5 M to N relationships. They are *watches*, *visits*, *displays*, *participates_in* and *holds*. Create a new table for each of them. Set primary keys and foreign keys respectively.

Step 7: Multi-valued Attributes

There are 2 multi-valued attributes. They are Award and Review of SCI_FI_MOVIE. Create a new table for each of them. Set primary keys and foreign keys respectively.

Step 8: N-Ary Relationship

There aren’t any.

Final Solution:

A9_VISITOR (Identifier, FirstName, LastName, Gender, Birthdate, Grade, Type)

A9_MUSEUM_CINEMA (CinemaName, ScreenType, Location, Capacity)

A9_SCI_FI_MOVIE (MovieTitle, Director, Price, Content, ReleaseDate, Runtime, Language, CinemaName)

FK: A9_SCI_FI_MOVIE.CinemaName -> A9_MUSEUM_CINEMA.CinemaName

A9_EXHIBITION (ExhibitionName, Theme, Price, Description, Location, StartDate, EndDate, Type)

A9_OBJECT (ObjectID, ObjectName, UsingCondition, Introduction)

A9_CHILDREN_PARK (ParkName, Theme, Price, Location)

A9_ACTIVITY (ActivityName, Presenter, Introduction)

A9_MOVIE_AWARD (MovieTitle, Award)

FK: A9_MOVIE_AWARD.MovieTitle -> A9_SCI_FI_MOVIE.MovieTitle

A9_MOVIE_REVIEW (MovieTitle, Review)

FK: A9_MOVIE_REVIEW.MovieTitle -> A9_SCI_FI_MOVIE.MovieTitle

A9_WATCHES (<u>Identifier, MovieTitle, WatchTime</u>, Cost)

FK: A9_WATCHES.Identifier -> A9_VISITOR.Identifier

FK: A9_WATCHES.MovieTitle -> A9_SCI_FI_MOVIE.MovieTitle

A9_VISITS (<u>Identifier, ExhibitionName, VisitTime</u>, Cost)

FK: A9_VISITS.Identifier -> A9_VISITOR.Identifier

FK: A9_VISITS.ExhibitionName -> A9_EXHIBITION.ExhibitionName

A9_DISPLAYS (<u>ExhibitionName, ObjectID, ExhibitDate</u>)

FK: A9_DISPLAYS.ExhibitionName -> A9_EXHIBITION.ExhibitionName

FK: A9_DISPLAYS.ObjectID -> A9_OBJECT.ObjectID

A9_PARTICIPATES_IN (<u>Identifier, ParkName, ParticipateTime</u>, Cost)

FK: A9_PARTICIPATES_IN.Identifier -> A9_VISITOR.Identifier

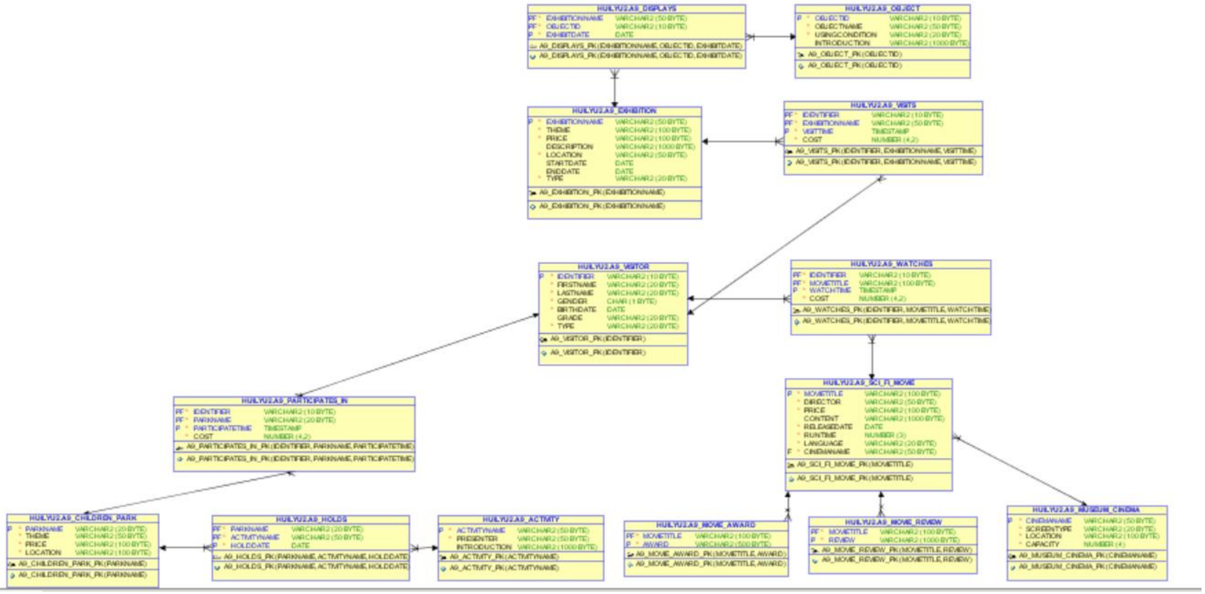
FK: A9_PARTICIPATES_IN.ParkName -> A9_CHILDREN_PARK.ParkName

A9_HOLDS (<u>ParkName, ActivityName, HoldDate</u>)

FK: A9_HOLDS.ParkName -> A9_CHILDREN_PARK.ParkName

FK: A9_HOLDS.ActivityName -> A9_ACTIVITY.ActivityName

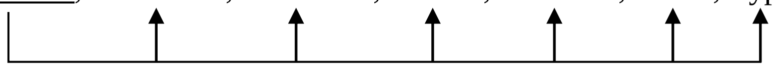
UML Diagram:


HUILYU2.A9_DISPLAYS
PF * EXHIBITIONNAME VARCHAR2 (50 BYTE)
PF * OBJECTID VARCHAR2 (10 BYTE)
P * EXHIBITDATE DATE
A9_DISPLAYS_PK (EXHIBITIONNAME, OBJECTID, EXHIBITDATE)
A9_DISPLAYS_PK (EXHIBITIONNAME, OBJECTID, EXHIBITDATE)
HUILYU2.A9_OBJECT
P * OBJECTID VARCHAR2 (10 BYTE)
* OBJECTNAME VARCHAR2 (50 BYTE)
* USINGCONDITION VARCHAR2 (20 BYTE)
INTRODUCTION VARCHAR2 (1000 BYTE)
A9_OBJECT_PK (OBJECTID)
A9_OBJECT_PK (OBJECTID)
HUILYU2.A9_EXHIBITION
P * EXHIBITIONNAME VARCHAR2 (50 BYTE)
* THEME VARCHAR2 (100 BYTE)
* PRICE VARCHAR2 (100 BYTE)
DESCRIPTION VARCHAR2 (1000 BYTE)
* LOCATION VARCHAR2 (50 BYTE)
STARTDATE DATE
ENDDATE DATE
* TYPE VARCHAR2 (20 BYTE)
A9_EXHIBITION_PK (EXHIBITIONNAME)
A9_EXHIBITION_PK (EXHIBITIONNAME)
HUILYU2.A9_VISITS
PF * IDENTIFIER VARCHAR2 (10 BYTE)
PF * EXHIBITIONNAME VARCHAR2 (50 BYTE)
P * VISITTIME TIMESTAMP
* COST NUMBER (4,2)
A9_VISITS_PK (IDENTIFIER, EXHIBITIONNAME, VISITTIME)
A9_VISITS_PK (IDENTIFIER, EXHIBITIONNAME, VISITTIME)
HUILYU2.A9_VISITOR
P * IDENTIFIER VARCHAR2 (10 BYTE)
* FIRSTNAME VARCHAR2 (20 BYTE)
* LASTNAME VARCHAR2 (20 BYTE)
* GENDER CHAR (1 BYTE)
* BIRTHDATE DATE
GRADE VARCHAR2 (20 BYTE)
* TYPE VARCHAR2 (20 BYTE)
A9_VISITOR_PK (IDENTIFIER)
A9_VISITOR_PK (IDENTIFIER)
HUILYU2.A9_WATCHES
PF * IDENTIFIER VARCHAR2 (10 BYTE)
PF * MOVIETITLE VARCHAR2 (100 BYTE)
P * WATCHTIME TIMESTAMP
* COST NUMBER (4,2)
A9_WATCHES_PK (IDENTIFIER, MOVIETITLE, WATCHTIME)
A9_WATCHES_PK (IDENTIFIER, MOVIETITLE, WATCHTIME)
HUILYU2.A9_SCI_FI_MOVIE
P * MOVIETITLE VARCHAR2 (100 BYTE)
* DIRECTOR VARCHAR2 (50 BYTE)
* PRICE VARCHAR2 (100 BYTE)
CONTENT VARCHAR2 (1000 BYTE)
* RELEASEDATE DATE
* RUNTIME NUMBER (3)
* LANGUAGE VARCHAR2 (20 BYTE)
F * CINEMANAME VARCHAR2 (50 BYTE)
A9_SCI_FI_MOVIE_PK (MOVIETITLE)
A9_SCI_FI_MOVIE_PK (MOVIETITLE)
HUILYU2.A9_PARTICIPATES_IN
PF * IDENTIFIER VARCHAR2 (10 BYTE)
PF * PARKNAME VARCHAR2 (20 BYTE)
P * PARTICIPATETIME TIMESTAMP
* COST NUMBER (4,2)
A9_PARTICIPATES_IN_PK (IDENTIFIER, PARKNAME, PARTICIPATETIME)
A9_PARTICIPATES_IN_PK (IDENTIFIER, PARKNAME, PARTICIPATETIME)
HUILYU2.A9_CHILDREN_PARK
P * PARKNAME VARCHAR2 (20 BYTE)
* THEME VARCHAR2 (50 BYTE)
* PRICE VARCHAR2 (100 BYTE)
* LOCATION VARCHAR2 (100 BYTE)
A9_CHILDREN_PARK_PK (PARKNAME)
A9_CHILDREN_PARK_PK (PARKNAME)
HUILYU2.A9_HOLDS
PF * PARKNAME VARCHAR2 (20 BYTE)
PF * ACTIVITYNAME VARCHAR2 (50 BYTE)
P * HOLDDATE DATE
A9_HOLDS_PK (PARKNAME, ACTIVITYNAME, HOLDDATE)
A9_HOLDS_PK (PARKNAME, ACTIVITYNAME, HOLDDATE)
HUILYU2.A9_ACTIVITY
P * ACTIVITYNAME VARCHAR2 (50 BYTE)
* PRESENTER VARCHAR2 (50 BYTE)
INTRODUCTION VARCHAR2 (1000 BYTE)
A9_ACTIVITY_PK (ACTIVITYNAME)
A9_ACTIVITY_PK (ACTIVITYNAME)
HUILYU2.A9_MOVIE_AWARD
PF * MOVIETITLE VARCHAR2 (100 BYTE)
P * AWARD VARCHAR2 (500 BYTE)
A9_MOVIE_AWARD_PK (MOVIETITLE, AWARD)
A9_MOVIE_AWARD_PK (MOVIETITLE, AWARD)
HUILYU2.A9_MOVIE_REVIEW
PF * MOVIETITLE VARCHAR2 (100 BYTE)
P * REVIEW VARCHAR2 (1000 BYTE)
A9_MOVIE_REVIEW_PK (MOVIETITLE, REVIEW)
A9_MOVIE_REVIEW_PK (MOVIETITLE, REVIEW)
HUILYU2.A9_MUSEUM_CINEMA
P * CINEMANAME VARCHAR2 (50 BYTE)
* SCREENTYPE VARCHAR2 (20 BYTE)
* LOCATION VARCHAR2 (100 BYTE)
* CAPACITY NUMBER (4)
A9_MUSEUM_CINEMA_PK (CINEMANAME)
A9_MUSEUM_CINEMA_PK (CINEMANAME)

Part 4: Functional Dependencies

(1)

A9_VISITOR (Identifier, FirstName, LastName, Gender, Birthdate, Grade, Type)



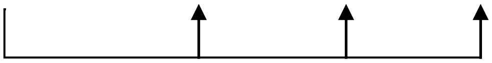
This table is in 2NF. There is only one attribute that composes the primary key. The primary key is Identifier. All of the other non-key attributes are fully functionally dependent by Identifier. In other words, all non-key attributes are fully functionally dependent by the whole primary key of the table instead of a part of the primary key.

This table is also in 3NF. All of the non-key attributes are fully functionally dependent by Identifier, the primary key. Every non-key attribute in the table does not depend on another non-key attribute. In other words, there is no transitive dependency between non-key attributes. (Birthdate does not determine Type because it is not absolutely right that all children go to school at the same age.)

In this table, Identifier determines FirstName, LastName, Gender, Birthdate, Grade and Type.

(2)

A9_MUSEUM_CINEMA (CinemaName, ScreenType, Location, Capacity)



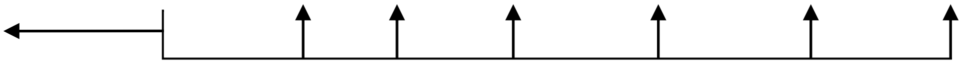
This table is in 2NF. There is only one attribute that composes the primary key. The primary key is CinemaName. All of the other non-key attributes are fully functionally dependent by CinemaName. In other words, all non-key attributes are fully functionally dependent by the whole primary key of the table instead of a part of the primary key.

This table is also in 3NF. All of the non-key attributes are fully functionally dependent by CinemaName, the primary key. Every non-key attribute in the table does not depend on another non-key attribute. In other words, there is no transitive dependency between non-key attributes. (ScreenType does not always determine Location and Capacity, because there may be some changes among several years in the science and technology museum.)

In this table, CinemaName determines ScreenType, Location and Capacity.

(3)

A9_SCI_FI_MOVIE (MovieTitle, Director, Price, Content, ReleaseDate, Runtime, Language, CinemaName)



This table is in 2NF. There is only one attribute that composes the primary key. The primary key is MovieTitle. All of the other non-key attributes are fully functionally dependent by MovieTitle.

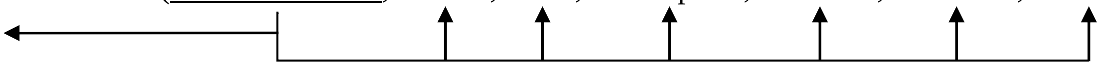
In other words, all non-key attributes are fully functionally dependent by the whole primary key of the table instead of a part of the primary key.

This table is also in 3NF. All of the non-key attributes are fully functionally dependent by MovieTitle, the primary key. Every non-key attribute in the table does not depend on another non-key attribute. In other words, there is no transitive dependency between non-key attributes.

In this table, MovieTitle determines Director, Price, Content, ReleaseDate, Runtime, Language and CinemaName.

(4)

A9_EXHIBITION (ExhibitionName, Theme, Price, Description, Location, StartDate, EndDate, Type)



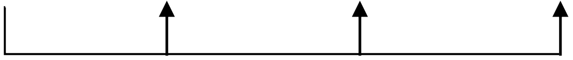
This table is in 2NF. There is only one attribute that composes the primary key. The primary key is ExhibitionName. All of the other non-key attributes are fully functionally dependent by ExhibitionName. In other words, all non-key attributes are fully functionally dependent by the whole primary key of the table instead of a part of the primary key.

This table is also in 3NF. All of the non-key attributes are fully functionally dependent by ExhibitionName, the primary key. Every non-key attribute in the table does not depend on another non-key attribute. In other words, there is no transitive dependency between non-key attributes. (Theme dose not determine Description because exhibitions with the same theme may have different descriptions. Also, Location does not determine Type because there may be some changes among several years.)

In this table, ExhibitionName determines Theme, Price, Description, Location, StartDate, EndDate and Type.

(5)

A9_OBJECT (ObjectID, ObjectName, UsingCondition, Introduction)



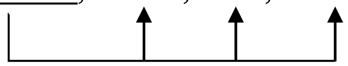
This table is in 2NF. There is only one attribute that composes the primary key. The primary key is ObjectID. All of the other non-key attributes are fully functionally dependent by ObjectID. In other words, all non-key attributes are fully functionally dependent by the whole primary key of the table instead of a part of the primary key.

This table is also in 3NF. All of the non-key attributes are fully functionally dependent by ObjectID, the primary key. Every non-key attribute in the table does not depend on another non-key attribute. In other words, there is no transitive dependency between non-key attributes. (ObjectName does not determine Introduction because objects with the same name may have different introductions.)

In this table, ObjectID determines ObjectName, UsingCondition and Introduction.

(6)

A9_CHILDREN_PARK (ParkName, Theme, Price, Location)



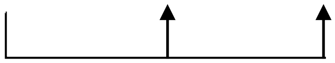
This table is in 2NF. There is only one attribute that composes the primary key. The primary key is ParkName. All of the other non-key attributes are fully functionally dependent by ParkName. In other words, all non-key attributes are fully functionally dependent by the whole primary key of the table instead of a part of the primary key.

This table is also in 3NF. All of the non-key attributes are fully functionally dependent by ParkName, the primary key. Every non-key attribute in the table does not depend on another non-key attribute. In other words, there is no transitive dependency between non-key attributes.

In this table, ParkName determines Theme, Price and Location.

(7)

A9_ACTIVITY (ActivityName, Presenter, Introduction)



This table is in 2NF. There is only one attribute that composes the primary key. The primary key is ActivityName. All of the other non-key attributes are fully functionally dependent by ActivityName. In other words, all non-key attributes are fully functionally dependent by the whole primary key of the table instead of a part of the primary key.

This table is also in 3NF. All of the non-key attributes are fully functionally dependent by ActivityName, the primary key. Every non-key attribute in the table does not depend on another non-key attribute. In other words, there is no transitive dependency between non-key attributes.

In this table, ActivityName determines Presenter and Introduction.

(8)

A9_MOVIE_AWARD (MovieTitle, Award)

This table is in 2NF. All of the attributes compose the primary key. The compound primary key is MovieTitle and Award. There is no non-key attribute in the table.

This table is also in 3NF. Since there is no non-key attribute in the table, there is no transitive dependency between non-key attributes.

(9)

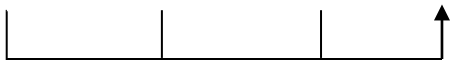
A9_MOVIE_REVIEW (MovieTitle, Review)

This table is in 2NF. All of the attributes compose the primary key. The compound primary key is MovieTitle and Review. There is no non-key attribute in the table.

This table is also in 3NF. Since there is no non-key attribute in the table, there is no transitive dependency between non-key attributes.

(10)

A9_WATCHES (Identifier, MovieTitle, WatchTime, Cost)



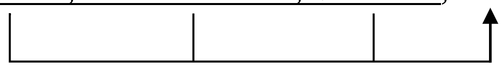
This table is in 2NF. Identifier, MovieTitle and WatchTime compose the primary key. The compound primary key is Identifier, MovieTitle and WatchTime. There is only one non-key attribute: Cost. Cost is fully functionally dependent by Identifier, MovieTitle and WatchTime. In other words, all non-key attributes are fully functionally dependent by the whole primary key of the table instead of a part of the primary key.

This table is also in 3NF. All of the non-key attributes are fully functionally dependent by Identifier, MovieTitle and WatchTime, the primary key. Since there is only one non-key attribute, there is no transitive dependency between non-key attributes.

In this table, Identifier, MovieTitle and WatchTime together determine Cost.

(11)

A9_VISITS (Identifier, ExhibitionName, VisitTime, Cost)



This table is in 2NF. Identifier, ExhibitionName and VisitTime compose the primary key. The compound primary key is Identifier, ExhibitionName and VisitTime. There is only one non-key attribute: Cost. Cost is fully functionally dependent by Identifier, ExhibitionName and VisitTime. In other words, all non-key attributes are fully functionally dependent by the whole primary key of the table instead of a part of the primary key.

This table is also in 3NF. All of the non-key attributes are fully functionally dependent by Identifier, ExhibitionName and VisitTime, the primary key. Since there is only one non-key attribute, there is no transitive dependency between non-key attributes.

In this table, Identifier, ExhibitionName and VisitTime together determine Cost.

(12)

A9_DISPLAYS (ExhibitionName, ObjectID, ExhibitDate)

This table is in 2NF. All of the attributes compose the primary key. The compound primary key is ExhibitionName, ObjectID and ExhibitDate. There is no non-key attribute in the table.

This table is also in 3NF. Since there is no non-key attribute in the table, there is no transitive dependency between non-key attributes.

(13)

A9_PARTICIPATES_IN (Identifier, ParkName, ParticipateTime, Cost)

This table is in 2NF. Identifier, ParkName and ParticipateTime compose the primary key. The compound primary key is Identifier, ParkName and ParticipateTime There is only one non-key attribute: Cost. Cost is fully functionally dependent by Identifier, ParkName and ParticipateTime. In other words, all non-key attributes are fully functionally dependent by the whole primary key of the table instead of a part of the primary key.

This table is also in 3NF. All of the non-key attributes are fully functionally dependent by Identifier, ParkName and ParticipateTime, the primary key. Since there is only one non-key attribute, there is no transitive dependency between non-key attributes.

In this table, Identifier, ParkName and ParticipateTime together determine Cost.

(14)

A9_HOLDS (ParkName, ActivityName, HoldDate)

This table is in 2NF. All of the attributes compose the primary key. The compound primary key is ParkName, ActivityName and HoldDate. There is no non-key attribute in the table.

This table is also in 3NF. Since there is no non-key attribute in the table, there is no transitive dependency between non-key attributes.

Part 5: Add Records

Part 6: Queries

Creating tables:

(1)

```
  CREATE TABLE "HUILYU2"."A9_ACTIVITY"
   (	"ACTIVITYNAME" VARCHAR2(50 BYTE) NOT NULL ENABLE,
	"PRESENTER" VARCHAR2(50 BYTE) NOT NULL ENABLE,
	"INTRODUCTION" VARCHAR2(1000 BYTE),
	 CONSTRAINT "A9_ACTIVITY_PK" PRIMARY KEY ("ACTIVITYNAME")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT)
  TABLESPACE "USERS"  ENABLE
   ) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
 NOCOMPRESS LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT)
  TABLESPACE "USERS" ;
```

(2)

```
  CREATE TABLE "HUILYU2"."A9_CHILDREN_PARK"
   (        "PARKNAME" VARCHAR2(20 BYTE) NOT NULL ENABLE,
            "THEME" VARCHAR2(50 BYTE) NOT NULL ENABLE,
            "PRICE" VARCHAR2(100 BYTE) NOT NULL ENABLE,
            "LOCATION" VARCHAR2(100 BYTE) NOT NULL ENABLE,
             CONSTRAINT "A9_CHILDREN_PARK_PK" PRIMARY KEY ("PARKNAME")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT)
  TABLESPACE "USERS"  ENABLE
   ) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
 NOCOMPRESS LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT)
  TABLESPACE "USERS" ;
```

(3)

```
  CREATE TABLE "HUILYU2"."A9_DISPLAYS"
   (	"EXHIBITIONNAME" VARCHAR2(50 BYTE) NOT NULL ENABLE,
	"OBJECTID" VARCHAR2(10 BYTE) NOT NULL ENABLE,
	"EXHIBITDATE" DATE NOT NULL ENABLE,
	 CONSTRAINT "A9_DISPLAYS_PK" PRIMARY KEY ("EXHIBITIONNAME", "OBJECTID", "EXHIBITDATE")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT)
  TABLESPACE "USERS"  ENABLE,
	 CONSTRAINT "A9_DISPLAYS_A9_EXHIBITION_FK1" FOREIGN KEY ("EXHIBITIONNAME")
	  REFERENCES "HUILYU2"."A9_EXHIBITION" ("EXHIBITIONNAME") ENABLE,
	 CONSTRAINT "A9_DISPLAYS_A9_OBJECT_FK1" FOREIGN KEY ("OBJECTID")
	  REFERENCES "HUILYU2"."A9_OBJECT" ("OBJECTID") ENABLE
   ) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
 NOCOMPRESS LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT)
  TABLESPACE "USERS" ;
```

(4)

```
  CREATE TABLE "HUILYU2"."A9_EXHIBITION"
   (	"EXHIBITIONNAME" VARCHAR2(50 BYTE) NOT NULL ENABLE,
	"THEME" VARCHAR2(100 BYTE) NOT NULL ENABLE,
	"PRICE" VARCHAR2(100 BYTE) NOT NULL ENABLE,
	"DESCRIPTION" VARCHAR2(1000 BYTE),
	"LOCATION" VARCHAR2(50 BYTE) NOT NULL ENABLE,
	"STARTDATE" DATE,
	"ENDDATE" DATE,
	"TYPE" VARCHAR2(20 BYTE) NOT NULL ENABLE,
	 CONSTRAINT "A9_EXHIBITION_PK" PRIMARY KEY ("EXHIBITIONNAME")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT)
  TABLESPACE "USERS"  ENABLE
   ) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
 NOCOMPRESS LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT)
  TABLESPACE "USERS" ;
```

(5)

```
  CREATE TABLE "HUILYU2"."A9_HOLDS"
   (	"PARKNAME" VARCHAR2(20 BYTE) NOT NULL ENABLE,
	"ACTIVITYNAME" VARCHAR2(50 BYTE) NOT NULL ENABLE,
	"HOLDDATE" DATE NOT NULL ENABLE,
	 CONSTRAINT "A9_HOLDS_PK" PRIMARY KEY ("PARKNAME",
"ACTIVITYNAME", "HOLDDATE")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT)
  TABLESPACE "USERS"  ENABLE,
         CONSTRAINT "A9_HOLDS_A9_ACTIVITY_FK1" FOREIGN KEY
("ACTIVITYNAME")
          REFERENCES "HUILYU2"."A9_ACTIVITY" ("ACTIVITYNAME") ENABLE,
         CONSTRAINT "A9_HOLDS_A9_CHILDREN_PARK_FK1" FOREIGN KEY
("PARKNAME")
          REFERENCES "HUILYU2"."A9_CHILDREN_PARK" ("PARKNAME") ENABLE
   ) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
 NOCOMPRESS LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT)
  TABLESPACE "USERS" ;
```

(6)

```
  CREATE TABLE "HUILYU2"."A9_MOVIE_AWARD"
   (	"MOVIETITLE" VARCHAR2(100 BYTE) NOT NULL ENABLE,
	"AWARD" VARCHAR2(500 BYTE) NOT NULL ENABLE,
	 CONSTRAINT "A9_MOVIE_AWARD_PK" PRIMARY KEY ("MOVIETITLE",
"AWARD")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT)
  TABLESPACE "USERS"  ENABLE,
	 CONSTRAINT "A9_MOVIE_AWARD_A9_SCI_FI__FK1" FOREIGN KEY
("MOVIETITLE")
	  REFERENCES "HUILYU2"."A9_SCI_FI_MOVIE" ("MOVIETITLE") ON DELETE
CASCADE ENABLE
   ) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
 NOCOMPRESS LOGGING
```

```
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT)
  TABLESPACE "USERS" ;
(7)
  CREATE TABLE "HUILYU2"."A9_MOVIE_REVIEW"
   (	"MOVIETITLE" VARCHAR2(100 BYTE) NOT NULL ENABLE,
	"REVIEW" VARCHAR2(1000 BYTE) NOT NULL ENABLE,
	 CONSTRAINT "A9_MOVIE_REVIEW_PK" PRIMARY KEY ("MOVIETITLE",
"REVIEW")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT)
  TABLESPACE "USERS"  ENABLE,
	 CONSTRAINT "A9_MOVIE_REVIEW_A9_SCI_FI_FK1" FOREIGN KEY
("MOVIETITLE")
	  REFERENCES "HUILYU2"."A9_SCI_FI_MOVIE" ("MOVIETITLE") ON DELETE
CASCADE ENABLE
   ) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
 NOCOMPRESS LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT)
  TABLESPACE "USERS" ;
(8)
  CREATE TABLE "HUILYU2"."A9_MUSEUM_CINEMA"
   (	"CINEMANAME" VARCHAR2(50 BYTE) NOT NULL ENABLE,
	"SCREENTYPE" VARCHAR2(20 BYTE) NOT NULL ENABLE,
	"LOCATION" VARCHAR2(100 BYTE) NOT NULL ENABLE,
	"CAPACITY" NUMBER(4,0) NOT NULL ENABLE,
	 CONSTRAINT "A9_MUSEUM_CINEMA_PK" PRIMARY KEY ("CINEMANAME")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT)
  TABLESPACE "USERS"  ENABLE
   ) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
 NOCOMPRESS LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT)
  TABLESPACE "USERS" ;
```

(9)

```
  CREATE TABLE "HUILYU2"."A9_OBJECT"
   (	"OBJECTID" VARCHAR2(10 BYTE) NOT NULL ENABLE,
	"OBJECTNAME" VARCHAR2(50 BYTE) NOT NULL ENABLE,
	"USINGCONDITION" VARCHAR2(20 BYTE) NOT NULL ENABLE,
	"INTRODUCTION" VARCHAR2(1000 BYTE),
	 CONSTRAINT "A9_OBJECT_PK" PRIMARY KEY ("OBJECTID")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT)
  TABLESPACE "USERS"  ENABLE
   ) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
 NOCOMPRESS LOGGING
```

```
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT)
  TABLESPACE "USERS" ;
```

(10)

```
  CREATE TABLE "HUILYU2"."A9_PARTICIPATES_IN"
   (	"IDENTIFIER" VARCHAR2(10 BYTE) NOT NULL ENABLE,
	"PARKNAME" VARCHAR2(20 BYTE) NOT NULL ENABLE,
	"PARTICIPATETIME" TIMESTAMP (0) NOT NULL ENABLE,
	"COST" NUMBER(4,2) NOT NULL ENABLE,
	 CONSTRAINT "A9_PARTICIPATES_IN_PK" PRIMARY KEY ("IDENTIFIER", "PARKNAME", "PARTICIPATETIME")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT)
  TABLESPACE "USERS"  ENABLE,
	 CONSTRAINT "A9_PARTICIPATES_IN_A9_CHI_FK1" FOREIGN KEY ("PARKNAME")
	  REFERENCES "HUILYU2"."A9_CHILDREN_PARK" ("PARKNAME") ENABLE,
	 CONSTRAINT "A9_PARTICIPATES_IN_A9_VIS_FK1" FOREIGN KEY ("IDENTIFIER")
	  REFERENCES "HUILYU2"."A9_VISITOR" ("IDENTIFIER") ENABLE
   ) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
 NOCOMPRESS LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT)
  TABLESPACE "USERS" ;
```

(11)

```
  CREATE TABLE "HUILYU2"."A9_SCI_FI_MOVIE"
   (	"MOVIETITLE" VARCHAR2(100 BYTE) NOT NULL ENABLE,
	"DIRECTOR" VARCHAR2(50 BYTE) NOT NULL ENABLE,
	"PRICE" VARCHAR2(100 BYTE) NOT NULL ENABLE,
	"CONTENT" VARCHAR2(1000 BYTE),
	"RELEASEDATE" DATE NOT NULL ENABLE,
	"RUNTIME" NUMBER(3,0) NOT NULL ENABLE,
	"LANGUAGE" VARCHAR2(20 BYTE) NOT NULL ENABLE,
	"CINEMANAME" VARCHAR2(50 BYTE) NOT NULL ENABLE,
	 CONSTRAINT "A9_SCI_FI_MOVIE_PK" PRIMARY KEY ("MOVIETITLE")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT)
  TABLESPACE "USERS"  ENABLE,
	 CONSTRAINT "A9_SCI_FI_MOVIE_A9_MUSEUM_FK1" FOREIGN KEY
("CINEMANAME")
	  REFERENCES "HUILYU2"."A9_MUSEUM_CINEMA" ("CINEMANAME")
ENABLE
   ) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
 NOCOMPRESS LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT)
  TABLESPACE "USERS" ;
```

(12)

```
  CREATE TABLE "HUILYU2"."A9_VISITOR"
   (	"IDENTIFIER" VARCHAR2(10 BYTE) NOT NULL ENABLE,
        "FIRSTNAME" VARCHAR2(20 BYTE) NOT NULL ENABLE,
        "LASTNAME" VARCHAR2(20 BYTE) NOT NULL ENABLE,
        "GENDER" CHAR(1 BYTE) NOT NULL ENABLE,
        "BIRTHDATE" DATE NOT NULL ENABLE,
        "GRADE" VARCHAR2(20 BYTE),
        "TYPE" VARCHAR2(20 BYTE) NOT NULL ENABLE,
         CONSTRAINT "A9_VISITOR_PK" PRIMARY KEY ("IDENTIFIER")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT)
  TABLESPACE "USERS"  ENABLE
   ) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
 NOCOMPRESS LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT)
  TABLESPACE "USERS" ;
```

(13)

```
  CREATE TABLE "HUILYU2"."A9_VISITS"
   (    "IDENTIFIER" VARCHAR2(10 BYTE) NOT NULL ENABLE,
        "EXHIBITIONNAME" VARCHAR2(50 BYTE) NOT NULL ENABLE,
        "VISITTIME" TIMESTAMP (0) NOT NULL ENABLE,
        "COST" NUMBER(4,2) NOT NULL ENABLE,
         CONSTRAINT "A9_VISITS_PK" PRIMARY KEY ("IDENTIFIER",
"EXHIBITIONNAME", "VISITTIME")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT)
  TABLESPACE "USERS"  ENABLE,
	 CONSTRAINT "A9_VISITS_A9_EXHIBITION_FK1" FOREIGN KEY ("EXHIBITIONNAME")
	  REFERENCES "HUILYU2"."A9_EXHIBITION" ("EXHIBITIONNAME") ENABLE,
	 CONSTRAINT "A9_VISITS_A9_VISITOR_FK1" FOREIGN KEY ("IDENTIFIER")
	  REFERENCES "HUILYU2"."A9_VISITOR" ("IDENTIFIER") ENABLE
   ) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
 NOCOMPRESS LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT)
  TABLESPACE "USERS" ;
```

(14)

```
  CREATE TABLE "HUILYU2"."A9_WATCHES"
   (	"IDENTIFIER" VARCHAR2(10 BYTE) NOT NULL ENABLE,
	"MOVIETITLE" VARCHAR2(100 BYTE) NOT NULL ENABLE,
	"WATCHTIME" TIMESTAMP (0) NOT NULL ENABLE,
	"COST" NUMBER(4,2) NOT NULL ENABLE,
	 CONSTRAINT "A9_WATCHES_PK" PRIMARY KEY ("IDENTIFIER", "MOVIETITLE", "WATCHTIME")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT)
  TABLESPACE "USERS"  ENABLE,
	 CONSTRAINT "A9_WATCHES_A9_SCI_FI_MOVI_FK1" FOREIGN KEY ("MOVIETITLE")
```

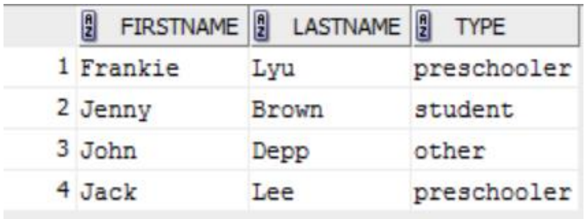
```
          REFERENCES "HUILYU2"."A9_SCI_FI_MOVIE" ("MOVIETITLE") ENABLE,
         CONSTRAINT "A9_WATCHES_A9_VISITOR_FK1" FOREIGN KEY ("IDENTIFIER")
          REFERENCES "HUILYU2"."A9_VISITOR" ("IDENTIFIER") ENABLE
  ) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
 NOCOMPRESS LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT)
  TABLESPACE "USERS" ;
```

Selecting records:

1. Select the distinct first names, last names and types (preschooler, student, other or dependent) of the visitors who have visited temporary exhibitions.

```
SELECT DISTINCT FIRSTNAME, LASTNAME, A9_VISITOR.TYPE
FROM A9_VISITOR
   JOIN A9_VISITS ON A9_VISITOR.IDENTIFIER = A9_VISITS.IDENTIFIER
   JOIN A9_EXHIBITION ON A9_VISITS.EXHIBITIONNAME = A9_EXHIBITION.EXHIBITIONNAME
WHERE A9_EXHIBITION.TYPE = 'Temporary Exhibition';
```

Result:



	FIRSTNAME	LASTNAME	TYPE
1	Frankie	Lyu	preschooler
2	Jenny	Brown	student
3	John	Depp	other
4	Jack	Lee	preschooler

2. What are the titles and prices of the movies which have more than two awards, and what are the capacities of the cinemas which play those movies?

SELECT A9_MOVIE_AWARD.MOVIETITLE, PRICE, CAPACITY, COUNT(AWARD)

FROM A9_SCI_FI_MOVIE, A9_MOVIE_AWARD, A9_MUSEUM_CINEMA

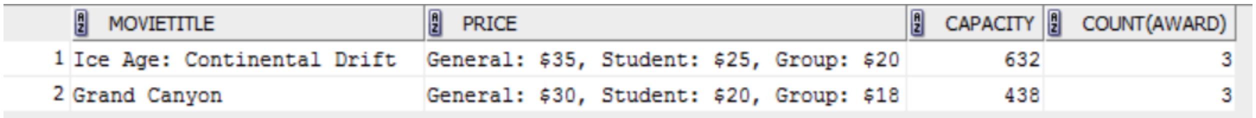
WHERE A9_MOVIE_AWARD.MOVIETITLE = A9_SCI_FI_MOVIE.MOVIETITLE

AND A9_SCI_FI_MOVIE.CINEMANAME = A9_MUSEUM_CINEMA.CINEMANAME

GROUP BY A9_MOVIE_AWARD.MOVIETITLE, PRICE, CAPACITY

HAVING COUNT(AWARD)>2;

Result:



	MOVIETITLE	PRICE	CAPACITY	COUNT(AWARD)
1	Ice Age: Continental Drift	General: $35, Student: $25, Group: $20	632	3
2	Grand Canyon	General: $30, Student: $20, Group: $18	438	3

3. What are the locations of exhibitions where objects in new using condition have been displayed? Present the objects' names and their corresponding display locations. Order by the exhibiting date in descending order.

SELECT OBJECTNAME, LOCATION, EXHIBITDATE

FROM A9_OBJECT, A9_EXHIBITION, A9_DISPLAYS

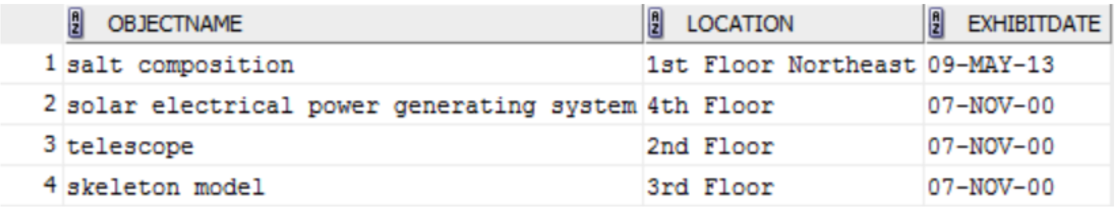
WHERE A9_OBJECT.OBJECTID = A9_DISPLAYS.OBJECTID

AND A9_DISPLAYS.EXHIBITIONNAME = A9_EXHIBITION.EXHIBITIONNAME

AND USINGCONDITION = 'new'

ORDER BY EXHIBITDATE DESC;

Result:

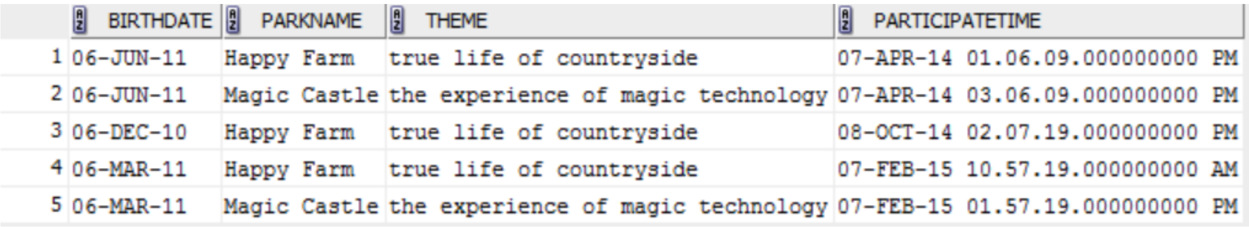


	OBJECTNAME	LOCATION	EXHIBITDATE
1	salt composition	1st Floor Northeast	09-MAY-13
2	solar electrical power generating system	4th Floor	07-NOV-00
3	telescope	2nd Floor	07-NOV-00
4	skeleton model	3rd Floor	07-NOV-00

4. List the birth dates of preschoolers (visitors) who have participated in children parks. Simultaneously, show the names and themes of the parks that they participated in. Order by the participating time in ascending order.

```
SELECT BIRTHDATE, A9_CHILDREN_PARK.PARKNAME, THEME, PARTICIPATETIME

FROM A9_VISITOR

   JOIN A9_PARTICIPATES_IN ON A9_VISITOR.IDENTIFIER = A9_PARTICIPATES_IN.IDENTIFIER

   JOIN A9_CHILDREN_PARK ON A9_CHILDREN_PARK.PARKNAME = A9_PARTICIPATES_IN.PARKNAME

ORDER BY PARTICIPATETIME;
```

Result:

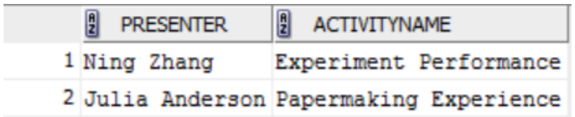


	BIRTHDATE	PARKNAME	THEME	PARTICIPATETIME
1	06-JUN-11	Happy Farm	true life of countryside	07-APR-14 01.06.09.000000000 PM
2	06-JUN-11	Magic Castle	the experience of magic technology	07-APR-14 03.06.09.000000000 PM
3	06-DEC-10	Happy Farm	true life of countryside	08-OCT-14 02.07.19.000000000 PM
4	06-MAR-11	Happy Farm	true life of countryside	07-FEB-15 10.57.19.000000000 AM
5	06-MAR-11	Magic Castle	the experience of magic technology	07-FEB-15 01.57.19.000000000 PM

5. What are the presenters who lead the activities which are held in 1st Floor Southeast Rm 106? Show the names of the activities that they lead.

```
SELECT PRESENTER, A9_ACTIVITY.ACTIVITYNAME

FROM A9_ACTIVITY, A9_CHILDREN_PARK, A9_HOLDS

WHERE A9_ACTIVITY.ACTIVITYNAME = A9_HOLDS.ACTIVITYNAME

    AND A9_HOLDS.PARKNAME = A9_CHILDREN_PARK.PARKNAME

    AND LOCATION = '1st Floor Southeast Rm 106';
```

Result:



	PRESENTER	ACTIVITYNAME
1	Ning Zhang	Experiment Performance
2	Julia Anderson	Papermaking Experience