

Homework 3 Solution

CS482 – Fall 2006

October 1, 2006

Problem 1. [Exercise 3.5] Looking at the data that happens to be stored in the tables for a particular application at some particular time, explain whether or not you can tell

- a. What the key constraints for the tables are

Answer: No, in one particular instance of the table, a particular attribute might uniquely identify the rows (and thus appear to be a key), but in other instances it might not.

- b. Whether or not a particular attribute forms a key for a particular table

Answer: You cannot tell whether a particular attribute is a key (for the same reason as in (a)), but you can sometimes tell that a particular attribute cannot form a key by itself (if two distinct rows have the same value over that attribute).

- c. What the integrity constraints for the application are

Answer: No. Again, one cannot determine integrity constraints just by looking at database instances.

- d. Whether or not a particular set of integrity constraints is satisfied

Answer: Yes. Simply check if every constraint in the set is satisfied in the given instance.

Problem 2. [Exercise 3.11]

```
-- Hw3, Ex 3.11
drop table HW3_RENTALS
/
drop table HW3_PAYMENTS
/
drop table HW3_UNIT
/
drop table HW3_PROPERTY
/
```

```
drop table HW3_CUSTOMER
/
```

```
PROMPT Creating Table 'HW3_PROPERTY'
CREATE TABLE HW3_PROPERTY
  (ID INTEGER NOT NULL
  ,ADDRESS VARCHAR2(200) NOT NULL
  ,NUMOFUNITS INTEGER DEFAULT 0 NOT NULL
  ,PRIMARY KEY(ID)
  )
/
```

```
PROMPT Creating Table 'HW3_CUSTOMER'
CREATE TABLE HW3_CUSTOMER
  (ID INTEGER NOT NULL
  ,HOME_ADDRESS VARCHAR2(200) NOT NULL
  ,PHONE INTEGER
  ,PRIMARY KEY(ID)
  )
/
```

```
PROMPT Creating Table 'HW3_UNIT'
CREATE TABLE HW3_UNIT
  (APARTNUM INTEGER NOT NULL
  ,PROPERTY_ID INTEGER NOT NULL
  ,RENTAL_PRICE NUMBER(6,2) NOT NULL
  ,UNIT_SIZE NUMBER(6,2)
  ,PRIMARY KEY(APARTNUM,PROPERTY_ID)
  ,FOREIGN KEY(PROPERTY_ID) REFERENCES HW3_PROPERTY(ID)
  )
/
```

```
PROMPT Creating Table 'HW3_PAYMENTS'
CREATE TABLE HW3_PAYMENTS
  (PAYMENT_ID INTEGER NOT NULL
  ,CUSTOMER_ID INTEGER NOT NULL
  ,UNIT_APARTNUM INTEGER NOT NULL
  ,PROPERTY_ID INTEGER NOT NULL
  ,AMOUNT NUMBER(6,2) NOT NULL
  ,PAYDATE DATE NOT NULL
  ,PRIMARY KEY(PAYMENT_ID)
  ,FOREIGN KEY(CUSTOMER_ID) REFERENCES HW3_CUSTOMER(ID)
  ,FOREIGN KEY(UNIT_APARTNUM,PROPERTY_ID) REFERENCES
    HW3_UNIT(APARTNUM,PROPERTY_ID)
  )
/
```

```

PROMPT Creating Table 'HW3_RENTALS'
CREATE TABLE HW3_RENTALS
  (RENTAL_ID INTEGER NOT NULL
  ,CUSTOMER_ID INTEGER NOT NULL
  ,PROPERTY_ID INTEGER NOT NULL
  ,UNIT_APARTNUM INTEGER NOT NULL
  ,RENT_DATE DATE NOT NULL
  ,RELEASE_DATE DATE
  ,PRIMARY KEY(RENTAL_ID)
  ,FOREIGN KEY(CUSTOMER_ID) REFERENCES HW3_CUSTOMER(ID)
  ,FOREIGN KEY(UNIT_APARTNUM,PROPERTY_ID) REFERENCES
      HW3_UNIT(APARTNUM,PROPERTY_ID)
  )
/

```

Note: All of the tables were created in the class database. To view the design of a table, you can use SQL Plus to connect to the class database CS482 and use the DESC command followed by the name of the table.

For example, if you want to view the columns of the table HW3_RENTALS, type the following command at the SQL Plus prompt:

```
SQL> desc HW3_RENTALS;
```

Problem 3. [Exercise 3.12]

```

drop table HW3_playedin
/
drop table HW3_movie
/
drop table HW3_actor
/
drop table HW3_studio
/

```

```

CREATE TABLE HW3_ACTOR
  (ID INTEGER NOT NULL
  ,NAME VARCHAR2(100) NOT NULL
  ,PRIMARY KEY(ID)
  )
/

```

```

CREATE TABLE HW3_STUDIO
  (ID INTEGER NOT NULL
  ,NAME VARCHAR2(100) NOT NULL

```

```

    ,PRIMARY KEY(ID)
)
/

CREATE TABLE HW3_MOVIE
(ID INTEGER NOT NULL
 ,NAME VARCHAR2(100) NOT NULL
 ,YEAR DATE
 ,STUDIO_ID INTEGER NOT NULL
 ,BUDGET NUMBER(10,2)
 ,PRIMARY KEY(ID)
 ,FOREIGN KEY(STUDIO_ID) REFERENCES HW3_STUDIO(ID)
)
/

CREATE TABLE HW3_PLAYEDIN
(ACTOR_ID INTEGER NOT NULL
 ,MOVIE_ID INTEGER NOT NULL
 ,PRIMARY KEY(ACTOR_ID,MOVIE_ID)
 ,FOREIGN KEY(ACTOR_ID) REFERENCES HW3_ACTOR(ID)
 ,FOREIGN KEY(MOVIE_ID) REFERENCES HW3_MOVIE(ID)
)
/

```

All of the tables were also created in the class database. See Problem 2 for how to view the design of the tables.