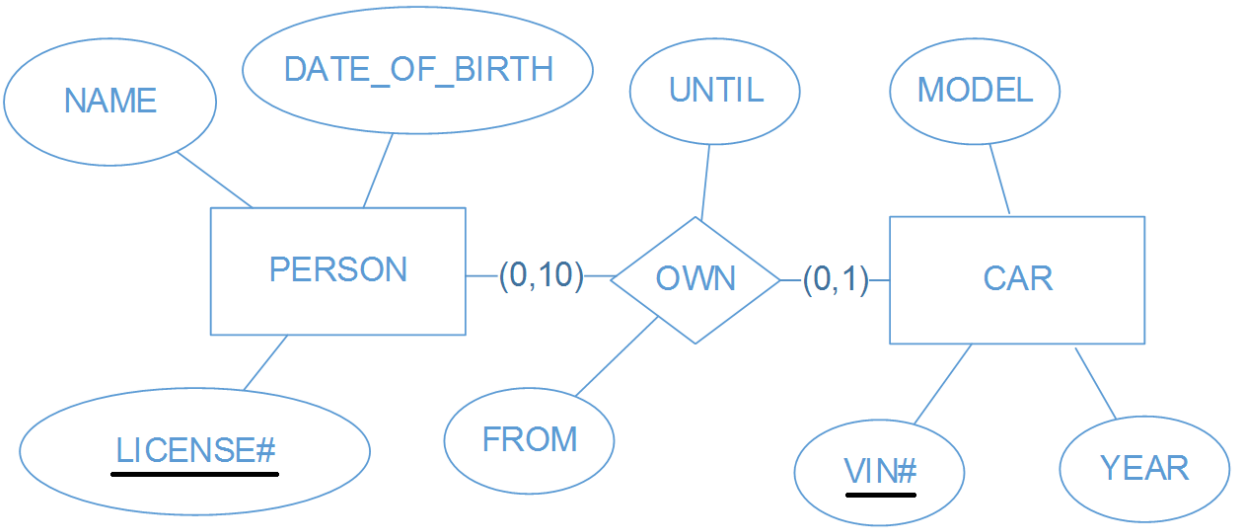
CS348: Introduction to Database Systems
(Fall 2014)

Assignment 4

Question 1. ER to Relational (10 marks)

For the scenario described herein, write SQL DDL statements (CREATE TABLE etc.) to create the **data model and constraints** that best capture the situation.



Question 2. (10 marks)

Let R be a relational schema with attributes {X, Q, Z, L, T, U} and let F be the set of functional dependencies F = {X → Q, X → Z, ZL → T, ZL → U, Q → T}.

a) What is the attribute closure of XL
b) Is XL a candidate key? Prove it.

Question 3. (10 marks)

Let R be a relational schema with attributes {V, W, X, Y, Z} and let F be the set of functional dependencies F = {V → WX, XY → Z, W → Y, Z → V }. Now suppose that we have the following decomposition of R:

R1= (V,W,X) and R2 = (V,Y,Z)

Prove that this decomposition is lossless.

Question 4. (10 marks)

Consider the following schedule, where $r_i(w)$ means that transaction i reads object v and $w_i(v)$ means that transaction i writes to object v.

$w_4(y)\ r_3(z)\ r_1(q)\ w_3(z)\ w_1(x)\ r_2(y)\ w_1(q)\ r_2(x)\ r_5(x)\ w_3(q)\ w_5(y)\ r_2(q)\ w_4(z)$

(a) (4 marks) List *all* pairs of conflicting operations in this schedule.

(b) (4 marks) Draw the conflict graph for this schedule.

(c) (2 marks) If the schedule is serializable, give an equivalent serial schedule. If it is not serializable, then explain why not.

Question 5. (5 marks)

Consider the following sequence of requests received by a database system that uses strict two-phase locking:

T_1: read x
T_3: read x
T_2: write x
T_2: read y
T_1: read y
T_3: read z
T_1: commit
T_4: write z
T_2: commit
T_5: write z
T_4: abort
T_3: commit
T_5: commit

Answer the following questions about this sequence of requests:

a) (2 marks) Which transactions get blocked?

b) (1 marks) Is there a deadlock? If yes between which transactions and data items?

c) (2 marks) Give an execution order that could result from these requests? Is this execution order serializable?

Question 6. (6 marks)

Consider a relation **mark** with schema **(<u>student, course, term</u>, grade)**. Assume that the instance of **mark** has 40,000 tuples covering 1000 students, 50 courses, and 10 terms, and that 20 records fit on each page (i.e., each disk access to the relation retrieves 20 records). Furthermore, assume that the data is clustered by **student**. Now consider the query

```
select count (distinct student)
from mark
where course = 'CS246'
```

and assume that the system returns the value 300 for this query.

(a) (3 marks) How many disk accesses would the system need to answer this query without using any index? Briefly explain why.

(b) (3 marks) Assume now that there is a B+ tree index on **mark (course, student)** having 5 levels (i.e., it takes 5 disk accesses to find a given course-student pair in the index), and that 50 course-student pairs fit on each leaf page (i.e., each disk access to a leaf page in the B+ tree retrieves 50 index records). Which of the following best approximates how many disk accesses would be needed to answer this query using the described index?

a. Less than 5
b. Between 5 and 15
c. Between 15 and 75
d. Between 75 and 350
e. More than 350

Briefly explain the algorithm you are assuming that the system uses to answer the query with this index.

Question 7 (10 marks)

Suppose that after a system failure, the transaction log looks as shown below (the log tail is at the bottom). A log entry (Ti, X, a, b) indicates that transaction Ti updated object X, changing its value from a to b. Describe what the database system must do to recover from the system failure. Indicate which objects must be modified, and in what order those modifications occur. Indicate which transactions are committed and which are aborted after the failure recovery is complete.

(T1,begin)
(T1,X,0,10)
(T2,begin)
(T2,Y,10,20)
(T3,begin)
(T1,commit)
(T3,X,10,400)
(T4,begin)
(T3,Z,0,100)
(T5,begin)
(T4,A,0,1)
(T6,begin)
(T4,abort)
(T5,A,0,2)
(T3,commit)
(T6,X,400,0)
(T5,commit)
(T6,A,2,3)