I. SQL lab.

Title: Kings

Description: The kings and queens of England are listed in a relation Kings(name,nickname,house,beginReign,endReign). Their name is unique, either using a Roman numeral to distinguish them, e.g., 'Edward I' or 'Edward II', or in a few cases using their nickname, e.g., 'Edward the Confessor'. The attribute nickname is an additional appelation, if they have one and it is not used in their name to distinguish them, e.g., 'The Unready'. The value of nickname is NULL if there is no nickname. The attribute house is the dynasty, e.g., 'Tudor'. Attribute beginReign and endReign are integers, the first and last years, respectively, that the king or queen was on the throne.
There is also a relation Parents(child,parent). Both attributes are the names of kings or queens, with the obvious connection that the first is a child of the second. Write the following queries:

1. *Who was king in the year 1000? Give the name and nickname.*
2. *Find all the pairs of kings or queens (A,B) such that A was the great grandchild of B.*
3. *Find the name and nickname of all kings or queens that have a nickname that does not begin with "The".*
4. *Find the names of those kings or queens that were the parent of two or more kings or queens. List each such person only once.*
5. *Find for each house the number of kings or queens of that house.*
6. *Several times in British history, kings or queens have deposed one another, so that their reigns overlapped. Find all such pairs, listing the pairs in both orders; i.e., list both (A,B) and (B,A). However, be careful not to list pairs A and B where the only overlap is that A's reign ended in the same year that B's began, or vice-versa.*

Solutions :

1. SELECT name, nickname FROM Kings WHERE beginReign <=1000 AND endReign >=1000

2. SELECT p1.child, p3.parent FROM Parents p1, Parents p2, Parents p3 WHERE p1.parent = p2.child AND p2.parent = p3.child

3. SELECT name, nickname FROM Kings WHERE nickname NOT LIKE 'The%'

4. SELECT parent FROM Parents GROUP by parent HAVING COUNT(*) > 1

5. SELECT house, COUNT(*) FROM Kings GROUP BY house

6. SELECT k1.name, k2.name FROM Kings k1, Kings k2 WHERE k1.name <> k2.name AND k1.beginReign < k2.endReign AND k2.beginReign < k1.endReign

II. Homework.

Question ID: 190 Here are declarations of two relations R and S:

```
CREATE TABLE S(
c INT PRIMARY KEY,
d INT
);
CREATE TABLE R(
a INT PRIMARY KEY,
b INT REFERENCES S(c)
);
```

R(a; b) currently contains the four tuples (0; 1); (7; 1); (3; 3), and (9; 9). S(c; d) currently contains the four tuples (1,4), (7,8), (9,4), and (3,1). Which of the following modi¯cations will not violate any constraint:

Correct choices:

Inserting (2, 3) into S
Inserting (4, 4) into S
Inserting (2, 1) into R
Inserting (4, 7) into R
Inserting (8, 9) into R
Inserting (8, 3) into R
Deleting (0, 1) from R
Deleting (7, 1) from R
Deleting (3, 3) from R
Deleting (9, 9) from R
Deleting (7, 8) from S

Incorrect choices:

Inserting (1, 5) into S
Inserting (7, 3) into S
Inserting (9, 2) into S
Inserting (3, 0) into S
Inserting (0, 7) into R
Inserting (7, 7) into R
Inserting (3, 9) into R
Inserting (9, 1) into R
Inserting (1, 5) into R
Inserting (8, 2) into R
Deleting (9, 4) from S
Deleting (1, 4) from S

Question ID 598: Here are declarations of two relations R and S:

```
CREATE TABLE S(
   c INT PRIMARY KEY,
   d INT
);
CREATE TABLE R(
   a INT PRIMARY KEY,
   b INT,
   CHECK(b IN (SELECT c FROM S))
);
```

R(a,b) currently contains the four tuples (0,4), (1,5), (2,4), and (3,5). S(c,d) currently contains the four tuples (2,10), (3,11), (4,12), and (5,13). As a result, certain insertions and deletions on S are illegal, as are certain updates or insertions on R. You should develop simple tests for illegal operations of these four kinds. Then, show your understanding by indicating which of the following modifications will not be rejected because of a constraint violation.

Correct choices

Correct Choice 1: Inserting (5,2) into R.
Correct Choice 2: Inserting (4,4) into R.
Correct Choice 3: Inserting (7,3) into R.
Correct Choice 4: Inserting (8,4) into R.
Correct Choice 5: Inserting (0,5) into S.
Correct Choice 6: Inserting (7,1) into S.
Correct Choice 7: Updating (0,4) in R to be (0,3).
Correct Choice 8: Updating (1,5) in R to be (1,2).
Correct Choice 9: Updating (2,4) in R to be (2,2).
Correct Choice 10: Updating (3,5) in R to be (3,3).
Correct Choice 11: Deleting (4,12) from S.
Correct Choice 12: Deleting (5,13) from S.

Incorrect choices

Incorrect Choice 1: Inserting (3,3) into R.

Choice Explanation: The key constraint for R prohibits inserting a tuple with R.a = 3, since one such tuple already exists in R. See Section 2.3.6 (p. 34).
Incorrect Choice 2: Inserting (1,4) into R.

Choice Explanation: The key constraint for R prohibits inserting a tuple with R.a = 1, since one such tuple already exists in R. See Section 2.3.6 (p. 34).
Incorrect Choice 3: Inserting (4,6) into R.

Choice Explanation: The CHECK constraint prohibits us from inserting into R a tuple with R.b equal to a value, such as 6, that is not currently a value of S.c. See Section 7.2.2 (p. 320).
Incorrect Choice 4: Inserting (5,0) into R.

Choice Explanation: The CHECK constraint prohibits us from inserting into R a tuple with R.b equal to a value, such as 0, that is not currently a value of S.c. See Section 7.2.2 (p. 320).
Incorrect Choice 5: Inserting (3,1) into S.

Choice Explanation: The key constraint for S prohibits inserting a tuple with S.c = 3, since one such tuple already exists in S. See Section 2.3.6 (p. 34).
Incorrect Choice 6: Inserting (4,13) into S.

Choice Explanation: The key constraint for S prohibits inserting a tuple with S.c = 4, since one such tuple already exists in S. See Section 2.3.6 (p. 34).
Incorrect Choice 7: Updating (0,4) in R to be (0,0).

Choice Explanation: The CHECK constraint prohibits us from updating R to have a tuple with R.b equal to a value, such as 0, that is not currently a value of S.c. See Section 7.2.2 (p. 320).
Incorrect Choice 8: Updating (1,5) in R to be (1,10).

Choice Explanation: The CHECK constraint prohibits us from updating R to have a tuple with R.b equal to a value, such as 10, that is not currently a value of S.c. See Section 7.2.2 (p. 320).
Incorrect Choice 9: Updating (2,4) in R to be (2,8).

Choice Explanation: The CHECK constraint prohibits us from updating R to have a tuple with R.b equal to a value, such as 8, that is not currently a value of S.c. See Section 7.2.2 (p. 320).
Incorrect Choice 10: Updating (3,5) in R to be (3,1).

Choice Explanation: The CHECK constraint prohibits us from updating R to have a tuple with R.b equal to a value, such as 1, that is not currently a value of S.c. See Section 7.2.2 (p. 320).

Question Explanation: The fact that c is a key of S says we cannot insert into S any tuple with first component 2, 3, 4, or 5. The fact that a is a key of R says we cannot insert into R any tuple with first component 0, 1, 2, or 3. The CHECK constraint says we cannot insert into R a tuple whose second component is not one of 2, 3, 4, or 5. Neither can we update a tuple to have a value of R.b that is none of 2, 3, 4, or 5.