

Name

CWID

# Homework Assignment 2

Due Date:  
October 17th, 2016  
12:30pm (noon)

## CS425 - Database Organization Results

---

*Please leave this empty!*

2.1  2.2

---

2.3  2.4  2.5  2.6  2.7  2.8  2.9  2.10

2.11  2.12

---

2.15  2.16  2.17  2.18  2.19  Sum

# Instructions

- Try to answer all the questions using what you have learned in class
- **When writing a query, write the query in a way that it would work over all possible database instances and not just for the given example instance!**
- Some questions are marked as bonus. You do not have to answer these questions to get full points for the assignment. However, you can get bonus points for these questions!
- **Please submit the homework electronically using blackboard**

Consider the following real estate database schema and example instance storing:

## buildingInfo

owner	neighbourhood	buildingNo	fromAptNo	toAptNo	owningsince
William	Prestwick Chase	455	5	10	1992
Sammy	Montenico	107	0	15	2001
George	Super Tech	114	10	15	1999
George	Prestwick Chase	377	5	10	1999
Cherry	The Hamilton	378	12	15	2003

## realtor

rName	neighbourhood	yrsOfcontract	neighbourhoodRates
Samuel	The Hamilton	5	2000
Monit	Montenico	2	2500
Adam	Prestwick Chase	6	2200
Samuel	Prestwick Chase	6	2200

## owner

ownerName	age	birthPlace
Cherry	48	Palm Beach
George	34	Miami
William	29	Tampa
Sammy	53	Orlando

## potentialcontract

tenant	buildingNo	realtor	aptNo
David	107	Monit	6
Hitesh	378	Samuel	13
Victor	455	Samuel	9
Garyr	107	Monit	14
Hitesh	455	Samuel	6

## tenant

aptNo	tenantName	buildingNo	livingSince	birthplace
8	Hitesh	377	2010	Tampa
9	Garyr	455	2011	Orlando
12	Victor	378	2011	Miami
7	David	107	2012	Orlando

### Hints:

- Attributes with black background form the primary key of an relation
- The attribute *owner* of relation *buildingInfo* is a foreign key to relation *owner* .
- The attribute *tenant* of relation *potentialContract* is a foreign key to relations *tenant*.
- The attribute *realtor* of relation *potentialContract* contains realtors, who deals and will deal on potential contract with tenants. However, it is not a foreign key to relation *realtor*, because the primary key of that relation also includes neighbourhood information.

- Assume that a building can be uniquely identified by *buildingNo*.

## Part 2.1 SQL DDL (Total: 14 Points)

### Question 2.1.1 (7 Points)

Write an SQL statement that creates a new table *newrental* that stores the *owner* and the owner's *buildingNo* and *aptNo* assigned to a *tenant*. Furthermore, we also want to store *neighbourhoodRates* in dollars for every new rental contract. The combination of owner, building no., apt no. and tenant uniquely identifies an new rental contract. For each contract make sure that the *neighbourhoodRate* for this contract is bigger than 1500 and smaller than 3000 dollar.

### Solution

```
CREATE TABLE newrental (  
    ownername VARCHAR(20),  
    buildingNo NUMBER,  
    aptNo NUMBER,  
    tenant VARCHAR(20),  
    neighbourrates NUMBER NOT NULL CHECK(neighbourrates BETWEEN 1501 AND 2999),  
    PRIMARY KEY (ownername, buildingNo, aptNo, tenant),  
    FOREIGN KEY (ownername) REFERENCES owner,  
);
```

optional additional foreign keys that are accepted:

```
FOREIGN KEY (tenantname) REFERENCES tenant,  
FOREIGN KEY (tenantname, buildingNo) REFERENCES potentialcontract
```

### Question 2.1.2 (7 Points)

Write an SQL statement that creates a table *dealsFor* that records which realtor deals buildings of which owner. For each such relationship between owners and realtors we record a commission for the realtor. A realtor may deal/sell buildings for several owners (and obviously a owner can employ multiple realtors).

```
CREATE TABLE dealsFor (  
  ownername VARCHAR2(20),  
  realtorname VARCHAR2(20),  
  commission NUMBER(8,0),  
  PRIMARY KEY (ownername, realtorname),  
  FOREIGN KEY ownername REFERENCES owner  
);
```

Note that realtor FK does not work, because neighbourhood is part of the PK.

## Part 2.2 SQL Queries (Total: 56 + 10 BONUS Points)

### Question 2.2.1 (5 Points)

Write an SQL query that returns the owner names and building no. of the buildings they own for all owners whose birthplace is Tampa or Orlando and who owns at least one a building in "The Hamilton" neighbourhood.

### Solution

```
SELECT b.owner , b.buildingNo
FROM buildingInfo b, owner o
WHERE o.ownername = b.owner
      AND (o.birthplace = 'Tampa' or o.birthplace = 'Orlando')
      AND b.owner IN (SELECT owner
                     FROM buildingInfo i
                     WHERE i.neighbourhood = 'The_Hamilton');
```

### Question 2.2.2 (5 Points)

Write an SQL query that returns bulding no. and tenant name for tentants that live in an apartment with apt. no between 8 and 11.

### Solution

```
SELECT buildingNo , tenantname
FROM tenant
WHERE aptno > 8 AND aptno < 11;
```

### Question 2.2.3 (7 Points)

Write an SQL query that counts the number of pairs of owners and tenants where all of the following conditions apply: 1) each tenant should only be paired with the owner of the building they live in, 2) the owner and tenant have the same birthplace, and 3) the tenant has been living at his/her current residence for more than 4 years.

### Solution

```
SELECT count(*)
FROM (SELECT DISTINCT b.owner , t.tenantname
      FROM buildingInfo b, tenant t, owner o
      WHERE b.buildingNo = t.buildingNo
           AND t.birthplace = o.birthplace
           AND o.ownername = b.owner
           AND to_char(sysdate, 'yyyy') - t.livingsince > 4) pairs
```

### Question 2.2.4 (7 Points)

Write an SQL query that returns all birthplaces for which the average age of owners for that place is above 40.

### Solution



```
SELECT birthplace
FROM owner
GROUP BY birthplace
HAVING avg(AGE) > 40;
```

### Question 2.2.5 (7 Points)

Write an SQL query that returns the owner-tenant pairs who are dealt by same realtor. For example, realtor Samuel (realtor table) deals in neighbourhood *The Hamilton*. Cherry owns a building in same neighbourhood (*The Hamilton*) and also tenant *Victor* living in buildingNo 377 of same neighbourhood *The Hamilton*. Thus, the pair Cherry and Victor would be in the result.

#### Solution

```
SELECT DISTINCT b.owner, t.tenantname
FROM buildingInfo b, tenant t
WHERE b.buildingNo = t.buildingNo;
```

or based on the correct understanding:

```
SELECT DISTINCT owner, tenantname
FROM (SELECT rName, tenantName
      FROM tenant t, realtor r, buildingInfo i
      WHERE t.buildingNo = i.buildingNo
            AND i.neighbourhood = r.neighbourhood) t
NATURAL JOIN
(SELECT rName, owner
 FROM buildingInfo i, realtor r
 WHERE i.neighbourhood = r.neighbourhood o
```

### Question 2.2.6 (8 Points)

Write an SQL query that returns potential tenants (names) and realtors (names) with the buildingNo and aptNo if the contract of the tenant currently occupying the apartment (if any) has ended. To determine whether a contract has expired, assume that the realtor mentioned in the potential contract is the same as the realtor that dealt with the current occupant of the apartment. Thus, you can use the value of attribute yrsOfcontract for this realtor for the neighborhood a building is located in together with livingSince to determine whether the contract of the current owner has expired. Hint: you can use SQL function `CURRENT_DATE` to get the current date and from that the year or alternatively assume that the current year is 2016.

### Solution

```
SELECT p.tenant, p.buildingno, p.realtor, p.aptNo
FROM potentialcontract p, tenant t, buildinginfo b, realtor r
WHERE p.buildingNo = t.buildingNo
      AND t.buildingNo = b.buildingNo
      AND t.aptNo = p.aptNo
      AND b.neighbourhood = r.neighbourhood
      AND r.rName = p.realtor
      AND to_char(sysdate, 'yyyy') - t.livingsince >= r.yrsofcontract
UNION
SELECT p.tenant, p.buildingno, p.realtor, p.aptNo
FROM potentialcontract p
WHERE NOT EXISTS (SELECT *
                  FROM tenant t
                  WHERE t.buildingNo = p.buildingNo
                        AND p.aptNo = t.aptNo);
```

### Question 2.2.7 (9 Points)

Write an SQL query that returns building numbers and apartment number(s) for the apartments which are available for rent, i.e., which are stored in the buildingInfo table, but are currently not occupied **and in tenant table, whose contract has expired.**

#### Solution

```
WITH aptNum AS (  
    SELECT level AS aptNo  
    FROM dual  
    CONNECT BY LEVEL <= (SELECT MAX(toAptNo)  
                          FROM buildingInfo)  
               AND LEVEL >= (SELECT MIN(fromAptNo)  
                              FROM buildingInfo)  
),  
unrollBInfo AS (  
    SELECT buildingNo, aptNo  
    FROM buildingInfo i, aptNum a  
    WHERE i.fromAptNo <= aptNo AND i.toAptNo >= aptNo  
)  
SELECT buildingNo, aptNo  
FROM unrollBInfo b  
WHERE NOT EXISTS (SELECT *  
                  FROM tenant t  
                  WHERE b.buildingNo = t.buildingNo AND b.aptNo = t.aptNo)  
ORDER BY buildingNo, aptNo;
```

The **ORDER BY** is optional.

### Question 2.2.8 (8 Points)

Write an SQL query which returns the name(s) of the tenants who are living in the neighbourhood with the highest neighbourhood rate.

#### Solution

```
SELECT t.tenantName
FROM tenant t, buildingInfo b, realtor r
WHERE t.buildingNo = b.buildingNo
      AND b.neighbourhood = r.neighbourhood
      AND r.neighbourhoodRates = (SELECT max(neighbourhoodRates)
                                  FROM realtor);
```

### Question 2.2.9 BONUS (5 Points)

Suppose the apartments in the potential contract table have all become unavailable, write an SQL query that displays for every potential tenant (person mentioned in potentialcontract) all available apartments by the same owner as the apartment the tenant was interested in. If no apartments of the owner are available, return all available apartments in the same neighbourhood.

#### Solution

```
WITH aptNum AS (
  SELECT level AS aptNo
  FROM dual
  CONNECT BY LEVEL <= (SELECT MAX(toAptNo)
                       FROM buildingInfo)
             AND LEVEL >= (SELECT MIN(fromAptNo)
                           FROM buildingInfo)
),
unrollBInfo AS (
  SELECT buildingNo, aptNo, owner, neighbourhood
  FROM buildingInfo i, aptNum a
  WHERE i.fromAptNo <= aptNo AND i.toAptNo >= aptNo
),
availApt AS (
  SELECT buildingNo, aptNo, owner, neighbourhood
  FROM unrollBInfo b
  WHERE NOT EXISTS (SELECT *
                   FROM tenant t
                   WHERE b.buildingNo = t.buildingNo AND b.aptNo = t.aptNo)
         AND NOT EXISTS (SELECT *
                        FROM potentialcontract p
                        WHERE b.buildingNo = p.buildingNo AND b.aptNo = p.aptNo)
),
interestNeighOwner AS (
  SELECT tenant, owner, neighbourhood
  FROM potentialcontract p, buildingInfo b
  WHERE b.buildingNo = p.buildingNo
)
SELECT o.tenant, a.buildingNo, a.aptNo
FROM interestNeighOwner o, availApt a
WHERE o.owner = a.owner
UNION ALL
SELECT o.tenant, a.buildingNo, a.aptNo
FROM interestNeighOwner o, availApt a
WHERE o.neighbourhood = a.neighbourhood
      AND NOT EXISTS (SELECT *
                     FROM interestNeighOwner i
                     WHERE i.owner = a.owner);
```

### Question 2.2.10 BONUS (5 Points)

Write an SQL query that returns the average number of buildings available to rent by owners for every birthplace.

#### Solution

```
SELECT avg(b.coun), birthplace
FROM owner o, (SELECT count(*) as coun, owner
               FROM buildingInfo
               GROUP BY owner) b
WHERE o.ownername = b.owner
GROUP BY birthplace
```

## Part 2.3 SQL Updates (Total: 30 + 5 BONUS Points)

### Question 2.3.1 (7 Points)

Write an SQL operation that deletes all buildings that were owned for more than 18 years.

#### Solution

```
DELETE FROM buildingInfo
WHERE to_char(sysdate, 'yyyy') - owningsince > 18;
```

### Question 2.3.2 (8 Points)

Decrease the rate of all neighborhoods dealt by realtor Samuel and Adam by 2,000.

#### Solution

```
UPDATE realtor
SET neighbourhoodRates = neighbourhoodRates - 2000
WHERE rName = 'Samuel' OR rName = 'Adam';
```



### Question 2.3.3 (6 Points)

Insert a new neighbourhood *Capendon* handled by realtor *Monit*. Buildings in this neighbourhood are available for contracts of 4 years and at \$2700 rate.

#### Solution

```
INSERT INTO realtor
VALUES ('Monit', 'Capendon', 4, 2700);
```

### Question 2.3.4 (9 Points)

Update livingSince to 2009 in the tenant table for tenants living in building with building number 377 if their current livingSince year is 2011.

#### Solution

```
UPDATE tenant
SET livingsince = 2009
WHERE buildingno = 377 AND livingsince = 2011;
```

### Question 2.3.5 BONUS (5 Points)

Change the contract years for *The Hamilton* and *Prestwick Chase* neighbourhood. Contract for *The Hamilton* neighbourhood should be set to the ceiling of the average contract length for this neighbourhood plus 2 and Contract for *Prestwick Chase* neighbourhood should be set to the maximum contract length for this neighbourhood minus 3.

### Solution

```
UPDATE realtor
SET yrsOfcontract = (CASE
    WHEN neighbourhood = 'The_Hamilton' THEN
        (SELECT ceil(avg(yrsOfcontract)) + 2
         FROM realtor
         WHERE neighbourhood = 'The_Hamilton')
    WHEN neighbourhood = 'Prestwick_Chase' THEN
        (SELECT max(yrsOfcontract) - 3
         FROM realtor
         WHERE neighbourhood = 'Prestwick_Chase')
    ELSE yrsOfcontract
END)
WHERE neighbourhood = 'The_Hamilton' OR neighbourhood = 'Prestwick_Chase';
```