CSCI 403 - Database Management
Sample Midterm Exam

Instructions:

There are 20 multiple choice questions below, worth 5 points each. Circle **one** answer for each question.

Questions:

Miscellaneous topics.

1. A functional dependency on a relation schema is:
   (a) A fact which we assert to be true for any instance of data which can be put in the relation.
   (b) A transient property of the data at a particular snapshot in time.
   (c) A maximal set of attributes of the relation which cannot contain null values.
   (d) A set of attributes of the relation for which no two tuples are allowed to share the same values.

2. How does using a B-tree data structure reduce the cost of searching for a key value in a database?
   (a) B-tree indices allow the hard disk to spin faster, thus reducing latency.
   (b) B-trees put most recently searched keys on the outer tracks of the hard drive, where rotational latency is at its minimum.
   (c) Navigating a B-tree requires reading only a few blocks from disk before the key is found (or determined not to exist in the database).
   (d) It doesn't; B-trees are useful only for enforcing primary key constraints.

These questions concern the database described by the data model and schema on the last page of the exam.

3. Which constraint(s) would be violated if we deleted the tuple (102, 20, 'F', 48) from the `individual` relation?
   (a) Primary key on the `individual` relation.
   (b) Foreign key on the `individual` relation referencing `household(id)`.
   (c) Foreign key on the `individual_race_xref` relation referencing `individual(id)`.
   (d) No constraints would be violated.

4. Which constraints on the `individual_race_xref` relation would be violated if we inserted the tuple (151, 2, null) into it?
   (a) Primary key.
   (b) Foreign key referencing `individual(id)`.
   (c) Foreign key referencing `race(id)`.
   (d) No constraints would be violated.

5. For someone just learning about census data, which query will best tell them the possible types of household units?

   (a) `SELECT * FROM household;`

   (b) `SELECT DISTINCT unit_type FROM household;`

   (c) `SELECT COUNT(unit_type) FROM household GROUP BY unit_type;`

   (d) `SELECT AVG(unit_type) FROM household;`

6. Which query will best list gender, age, city, and state for everyone in the census database sorted by state and city?

   (a)
   ```
   SELECT i.gender, i.age, loc.city, loc.state
   FROM individual AS i, locality AS loc
   ORDER BY loc.state, loc.city;
   ```

   (b)
   ```
   SELECT i.gender, i.age, loc.city, loc.state
   FROM individual AS i, household AS h, locality AS loc
   WHERE i.household_id = h.id AND h.locality_id = loc.id
   ORDER BY loc.state, loc.city;
   ```

   (c)
   ```
   SELECT *
   FROM individual AS i, household AS h, locality AS loc, individual_race_xref AS x, race
   WHERE i.household_id = h.id AND h.locality_id = loc.id AND x.individual_id = i.id
   AND x.race_id = race.id
   ORDER BY loc.state, loc.city;
   ```

   (d) `SELECT * FROM * ORDER BY state, city;`

7. Which query will list gender and for everyone living in Golden, CO?

   (a)
   ```
   SELECT gender, age FROM individual WHERE household_id IN
     (SELECT id FROM household WHERE locality_id =
        (SELECT id FROM locality
        WHERE city = 'Golden' AND county = 'Jefferson' AND state = 'CO'));
   ```

   (b)
   ```
   SELECT i.gender, i.age FROM individual AS i, household AS h, locality AS loc
   WHERE i.household_id = h.id AND h.locality_id = loc.id
   AND loc.city = 'Golden' AND loc.county = 'Jefferson' AND loc.state = 'CO';
   ```

   (c)
   ```
   SELECT gender, age FROM individual WHERE household_id IN
     (SELECT h.id FROM household AS h, locality AS loc
     WHERE h.locality_id = loc.id
     AND loc.city = 'Golden' AND loc.county = 'Jefferson' AND loc.state = 'CO');
   ```

   (d) All of the above.

8. Which query will correctly tabulate the number of residents by household (excluding empty households)?

   (a) `SELECT COUNT(*) FROM individual GROUP BY individual;`

   (b) `SELECT household_id, COUNT(*) FROM individual GROUP BY household_id;`

   (c) `SELECT COUNT(*) FROM individual ORDER BY household_id;`

   (d) `SELECT COUNT FROM individual WHERE household_id IN household;`

9. Which query will list the Asian racial groups recognized by the census?

   (a) `SELECT description FROM race WHERE description LIKE 'asian%';`

   (b) `SELECT description FROM race WHERE description = 'asian*';`

   (c) `SELECT description FROM race WHERE 'asian' IN description;`

   (d) None of the above.

10. Which query will correctly list all of the households with more than five residents in a rental property?

   (a)
   ```
   SELECT h.id FROM household AS h, individual AS i
   WHERE i.household_id = h.id AND h.unit_type = 'renter_occupied'
   GROUP BY h.id HAVING COUNT(*) > 5;
   ```

   (b)
   ```
   SELECT h.id FROM household AS h, individual AS i
   WHERE i.household_id = h.id AND h.unit_type = 'renter_occupied'
   AND COUNT(*) > 5 GROUP BY h.id;
   ```

   (c)
   ```
   SELECT h.id FROM household AS h, individual AS i
   WHERE i.household_id = h.id AND h.unit_type = 'renter_occupied'
   GROUP BY h.id WHERE COUNT(*) > 5;
   ```

   (d) All of the above.

11. The following data is collected on an individual by a census taker:

Household id:	26
Gender:	M
Age:	23
Race:	Asian - Indian

   Assuming the next unassigned individual_id is 161, what is a correct sequence of SQL commands to add this individual to the database?

   (a)
   ```
   INSERT INTO individual_race_xref (individual_id, race_id, specific_name)
   VALUES (161, 4, null);
   INSERT INTO individual (id, household_id, gender, age) VALUES (161, 26, 'M', 23);
   ```

   (b)
   ```
   INSERT INTO individual (id, household_id, gender, age) VALUES (161, 26, 'M', 23);
   INSERT INTO individual_race_xref (individual_id, race_id, specific_name)
   VALUES (161, 4, null);
   ```

   (c)
   ```
   INSERT INTO individual AS i, individual_race_xref AS x
   (i.id, i.household_id, i.gender, i.age, x.individual_id, x.race_id)
   VALUES (161, 26, 'M', 23, 161, 4) WHERE x.individual_id = i.id;
   ```

   (d) All of the above.

12. Which query below could you use to gather information about just the individuals who identify with specific racial sub-groups?

   (a)
   ```
   SELECT DISTINCT i.id, i.age, i.gender
   FROM individual AS i, individual_race_xref AS x
   WHERE x.individual_id = i.id AND x.specific_name <> NULL;
   ```

   (b)
   ```
   SELECT id, age, gender FROM individual WHERE id NOT IN
   (SELECT individual_id FROM individual_race_xref WHERE specific_name IS NULL);
   ```

   (c)
   ```
   SELECT id, age, gender FROM individual WHERE id IN
   (SELECT individual_id FROM individual_race_xref WHERE specific_name IS NOT NULL);
   ```

   (d) All of the above.

13. Based on the data model in figure 1, which of the following statements about census data collection is most likely true?

    (a) The designers assume that individuals each identify with a single race.

    (b) The designers assume that racial groups are divided by local boundaries.

    (c) The designers assume that each locality contains exactly one household.

    (d) The designers assume that people reside in one home only.

14. Why might the designers of the census database have chosen to make `locality` a separate entity (and relation) rather than adding city, county, and state attributes to the household table?

    (a) Locality data does change, even if only occasionally; having this information logically separated eliminates the chance of modification anomalies.

    (b) Storing a numerical identifier in the household table takes up a lot less space than city, county, and state data.

    (c) The entry of data into the database will be more consistent and accurate if localities are looked up from a static table rather than being entered individually for each household.

    (d) All of the above.

15. It is discovered that due to a data entry error, all of the rental properties in Lakewood, CO were incorrectly assigned to Arvada, CO. Fortunately, the data for rental properties in Arvada has not yet been entered (although other Arvada data has been entered), which means one solution to the problem is to execute the query or queries:

    (a) `UPDATE locality SET city = 'Lakewood' WHERE city = 'Arvada' AND state = 'CO';`

    (b)
    ```
    UPDATE household SET locality_id =
    (SELECT id FROM locality WHERE city = 'Lakewood' AND state = 'CO')
    WHERE locality_id =
    (SELECT id FROM locality WHERE city = 'Arvada' AND state = 'CO')
    AND unit_type = 'renter occupied';
    ```

    (c)
    ```
    UPDATE household AS h, locality AS loc SET h.locality_id =
    (SELECT id FROM locality WHERE city = 'Lakewood' AND state = 'CO')
    WHERE h.locality_id = loc.id AND loc.city = 'Arvada' AND loc.state = 'CO';
    ```

    (d)
    ```
    INSERT INTO household SELECT id, locality_id + 1, unit_type FROM household
    WHERE locality_id = 1337 AND unit_type = 'renter occupied';
    DELETE FROM household WHERE locality_id = 1337 AND unit_type = 'renter occupied';
    ```

16. In the distant future, houses will be mounted on anti-gravity units, which will allow them to be moved relatively easily as the season changes, in order to avoid the worst effects of global warming. For the census, households will be required to declare all of the localities in which they have airspace allotments. What is the best way to adapt the census data model and relational schema?

    (a) No changes are necessary, the existing model is just fine.

    (b) The "located in" relationship simply needs to be changed from 1-to-many to be many-to-1 instead. The household relation will lose its `locality_id` attribute, and the locality table will gain a `household_id` attribute.

    (c) The "located in" relationship will change to a many-to-many relationship. The household relation can no longer contain a `locality_id` attribute. Instead, a cross-reference table containing locality id's and household id's with foreign key references back to locality and household, respectively, must be created to represent the relationship.

    (d) The locality and household entities must be merged into one. The resulting merged relation will have rows for every combination of household and locality as relevant.

These questions concern the relation and dependencies described below.

Relation `scientists`:

Attributes:	{Archimedes, Bohr, Copernicus, Darwin, Einstein, Faraday, Galileo}
Keys:	
	{Archimedes, Copernicus, Galileo}
	{Darwin}
Some Functional Dependencies:	
	{Archimedes, Copernicus, Galileo} $\rightarrow$ {Bohr, Darwin, Einstein, Faraday}
	{Darwin} $\rightarrow$ {Archimedes}
	{Bohr} $\rightarrow$ {Faraday}
	{Galileo} $\rightarrow$ {Copernicus, Einstein}
	{Einstein} $\rightarrow$ {Bohr}

17. Which of the functional dependencies below violate Boyce-Codd Normal Form (BCNF)?
    (a) {Darwin} $\rightarrow$ {Archimedes}
    (b) {Bohr} $\rightarrow$ {Faraday}
    (c) {Galileo} $\rightarrow$ {Copernicus, Einstein}
    (d) Both (b) and (c)

18. The set {Copernicus, Galileo, Einstein} is:
    (a) A superkey.
    (b) The closure of {Galileo}.
    (c) The closure of {Copernicus, Einstein}.
    (d) None of the above.

19. Which additional functional dependencies can we infer to exist?
    (a) {Galileo} $\rightarrow$ {Faraday}
    (b) {Bohr} $\rightarrow$ {Einstein}
    (c) {Archimedes, Copernicus} $\rightarrow$ {Galileo}
    (d) {Faraday} $\rightarrow$ {Einstein}

20. Which of the following would be a decomposition of `scientists` which would move the schema closer to BCNF?
    (a) R1 = {Bohr, Copernicus, Einstein, Faraday, Galileo}, R2 = {Archimedes, Darwin, Galileo}
    (b) R1 = {Archimedes, Copernicus, Darwin, Galileo}, R2 = {Bohr, Darwin, Einstein, Faraday}
    (c) R1 = {Copernicus, Einstein, Galileo}, R2 = {Archimedes, Bohr, Darwin, Faraday}
    (d) R1 = {Bohr, Faraday}, R2 = {Archimedes, Bohr, Copernicus, Einstein, Galileo}

Answer key: 1 - (a), 2 - (c), 3 - (c), 4 - (d), 5 - (b), 6 - (b), 7 - (d), 8 - (b), 9 - (a), 10 - (a), 11 - (b), 12 - (c), 13 - (d), 14 - (d), 15 - (b), 16 - (c), 17 - (d), 18 - (b), 19 - (a), 20 - (a)

This page intentionally left blank.

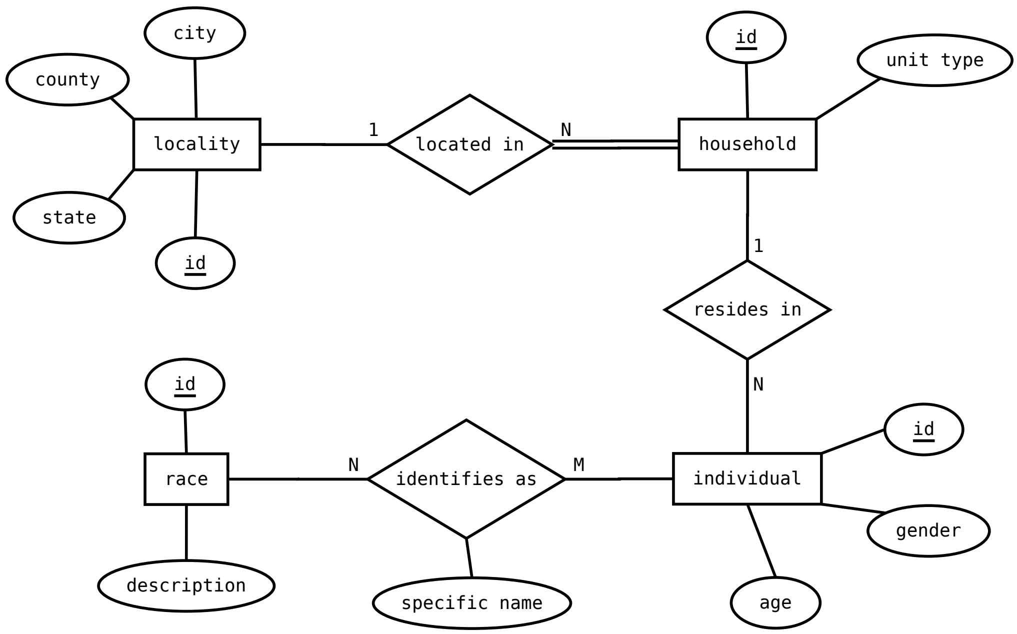

Figure 1: ERD for a census database

locality: Primary Key (id)

id	city	county	state
1337	Arvada	Jefferson	CO
1338	Lakewood	Jefferson	CO
1339	Golden	Jefferson	CO

household:
Primary Key (id)
Foreign Key (locality_id) on locality(id)

id	locality_id	unit_type
20	1337	renter occupied
21	1337	owner occupied
22	1337	owner occupied
23	1337	renter occupied
24	1337	vacant - for sale
25	1337	owner occupied

race: Primary Key (id)

id	description
1	white
2	black or african american
3	american indian or alaska native
4	asian - indian
5	asian - chinese
6	asian - filipino

individual:
Primary Key (id)
Foreign Key (household_id) on household(id)

id	household_id	gender	age
101	20	M	49
102	20	F	48
103	20	F	16
111	21	F	53
112	21	F	46
113	21	M	13
121	22	M	33
122	22	F	34
123	22	M	4
124	22	M	2
131	23	F	26
151	25	M	64
152	25	F	61

individual_race_xref:
Primary Key (individual_id, race_id)
Foreign Key (individual_id) on individual(id)
Foreign Key (race_id) on race(id)

individual_id	race_id	specific_name
101	1	null
102	2	null
103	1	null
111	3	Cheyenne
112	6	null
113	1	null
151	3	Arapaho
151	1	null
152	1	null

Figure 2: Relational schema and sample tuples for the same census database