This examination has 4 doublesided pages.

Check that you have a complete paper.

This is a closed book, closed notes exam. No books or other material may be used.

Answer all the questions on this paper.

Give very short but precise answers.

State any assumptions you make

Work fast and do the easy questions first. Leave some time to review your exam at the end.

Good Luck
All queries for this exam use the same schema as in some of the SQL tutorials:

- authors(au_id, au_lname, au_fname, phone, address, city, state, zip)
- titleauthors(au_id, title_id, au_ord, royaltyshare)
- sales(sonum, stor_id, ponum, sdate)
- salesdetails(sonum, qty_ordered, qty_shipped, title_id, date_shipped)
- editors(ed_id, ed_lname, ed_fname, ed_pos, phone, address, city, state, zip, ed_boss)
- titleditors(ed_id, title_id, ed_ord)
- titles(title_id, title, type, pub_id, price, advance, ytd_sales, contract, notes, pubdate)
- publishers(pub_id, pub_name, address, city, state)

The schema will be repeated on following pages for easy reference.

Foreign Keys are shown in the following diagram, where the referring attribute is marked by a + and the referencing attribute is marked by a ⇐ (e.g., au_id in titleauthors references au_id in authors)
The schema again:

authors
- au_id
- au_lname
- au_fname
- phone
- address
- city
- state
- zip

titleauthors
- au_id
- title_id
- au_ord
- royaltyshare

sales
- sonum
- stor_id
- ponum
- sdate

salesdetails
- sonum
- qty_order
- qty_shipped
- title_id
- date_shipped

editors
- ed_id
- ed_lname
- ed_fname
- ed_pos
- phone
- address
- city
- state
- zip

titleditors
- ed_id
- title_id
- ed_ord

titles
- title_id
- title
- type
- pub_id
- price
- advance
- ytd_sales
- contract
- notes
- pubdate

publishers
- pub_id
- pub_name
- address
- city
- state

1. a. **SQL**: Find the first name of all authors who are not editors. Remove duplicates and alphabetize.

   ```sql
   SELECT DISTINCT au_fname
   FROM authors a
   WHERE a.au_id NOT IN (SELECT e.ed_id FROM editors e)
   ORDER BY au_fname
   ```

```
AU_FNAME
-------------------
Abraham
Akiko
Albert
Ann
Anne
Burt
Chastity
Cheryl
Dick
Dirk
Heather
Innes
Johnson
Livia
Marjorie
Meander
Michael
Michel
Morningstar
Reginald
Sheryl
Stearns
Sylvi
```

23 rows selected.

Common errors:
- Comparing only first name or last name or something else of authors with that of editors (You were given marks if you compared both names or IDs)
- Forgetting “order by” or writing it incorrectly such as “Sort by” or not writing the attribute
- Forgetting distinct
- Trying to find editor_id through joining authors, titleauthors, titles and editors, by titleditors and titles
- Not using NOT EXISTS or NOT IN and comparing them simply comparing two tuples
The schema again:

- authors( au_id, au_lname, au_fname, phone, address, city, state, zip)
- titleauthors( au_id, title_id, au_ord, royaltyshare)
- sales( sonum, stor_id, ponum, sdate)
- salesdetails( sonum, qty_ordered, qty_shipped, title_id, date_shipped)
- editors( ed_id, ed_lname, ed_fname, ed_pos, phone, address, city, state, zip)
- titleditors( ed_id, title_id, ed_ord)
- titles( title_id, title, type, pub_id, price, advance, ytd_sales, contract, notes, pubdate)
- publishers( pub_id, pub_name, address, city, state)

b. SQL: “List the last names of all authors who have a letter 'k' in their last name?” If a last name occurs more than once, only list it once

Answer:
SELECT DISTINCT au_lname
FROM authors
WHERE au_lname LIKE '%k%' or au_lname LIKE '%K%'
Tuples:
Karsen
Locksley
Yokomoto

This is a question from the practice midterm from Spring 2010. It is also question 8j from the first SQL tutorial
Common error: You have to check for both the capital and lowercase K.
Also need to have distinct
The schema again:

```
authors( au_id, au_lname, au_fname, phone, address, city, state, zip)
titleauthors( au_id, title_id, au_ord, royaltyshare)
sales( sonum, stor_id, ponum, sdate)
salesdetails( sonum, qty_ordered, qty_shipped, title_id, date_shipped)
editors ( ed_id, ed_lname, ed_fname, ed_pos, phone, address, city, state, zip)
titleditors( ed_id, title_id, ed_ord)
titles( title_id, title, type, pub_id, price, advance, ytd_sales, contract, notes, pubdate)
publishers( pub_id, pub_name, address, city, state)
```

c: SQL. For each editor who has edited more than two books, return the last name of the editor and how many books she/he has edited

```
SELECT e.ed_lname, count(t.title_id)
FROM editors e, titleditors t
WHERE e.ed_id = t.ed_id
GROUP BY e.ed_id, e.ed_lname
HAVING count(title_id) > 2
```

<table>
<thead>
<tr>
<th>ED_LNAME</th>
<th>COUNT(T.TITLE_ID)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeLongue</td>
<td>6</td>
</tr>
<tr>
<td>Himmel</td>
<td>6</td>
</tr>
<tr>
<td>Rutherford-Hayes</td>
<td>5</td>
</tr>
<tr>
<td>Kaspchek</td>
<td>6</td>
</tr>
<tr>
<td>McCann</td>
<td>10</td>
</tr>
</tbody>
</table>

Common errors:
- You can’t just group by ed_id and then select the last name. Another way to do this is do a view/sub query to get the ed_id and then join to get the authors.
- You need to specify which ed_id to group by. Yes, they’re the same, but SQL isn’t that smart.
- You can’t just group by ed_lname because there may be more than one editor with the same last name.
- Renaming in select and using that within itself.
- No group by.
The schema again:
authors( au_id, au_lname, au_fname, phone, address, city, state, zip)
titleauthors( au_id, title_id, au_ord, royaltyshare)
sales( sonum, stor_id, ponum, sdate)
salesdetails( sonum, qty_ordered, qty_shipped, title_id, date_shipped)
editors ( ed_id, ed_lname, ed_fname, ed_pos, phone, address, city, state, zip)
titleditors( ed_id, title_id, ed_ord)
titles( title_id, title, type, pub_id, price, advance, ytd_sales, contract, notes, pubdate)
publishers( pub_id, pub_name, address, city, state)

d. Find the last name of the first author(s) of the book(s) that has had the most number of copies ordered. Remove duplicates.

```
CREATE VIEW total_sold(title_id, total_quantity) AS
SELECT title_id, SUM(qty_ordered)
FROM salesdetails
GROUP BY title_id

CREATE VIEW max_sold(title_id) AS
SELECT title_id
FROM total_sold t
WHERE t.total_quantity >= ALL ( 
    SELECT t2.total_quantity
    FROM total_sold t2
)

SELECT distinct (a.au_lname)
FROM authors a, max_sold m, titleauthors ta
WHERE a.au_id = ta.au_id AND ta.au_ord = 1 AND ta.title_id = m.title_id
```

Tuples:

```
AU_LNAME
----------------------------------------
Ringer
```

Common Errors:

- No grouping whatsoever (you need to add up all the orders – it’s not enough to find the number of books ordered in a single order)
- Wrong group by (e.g., author_id, other)
- Using aggregate functions in ‘where’ or ‘having’ clauses without their being part of a Boolean condition.
- Not using distinct
The schema again:

\begin{align*}
\text{authors(} & \text{au_id, au_lname, au_fname, phone, address, city, state, zip)} \\
\text{titleauthors(} & \text{au_id, title_id, au_ord, royaltyshare}) \\
\text{sales(} & \text{sonum, stor_id, ponum, sdate}) \\
\text{salesdetails(} & \text{sonum, qty_ordered, qty_shipped, title_id, date_shipped}) \\
\text{editors (} & \text{ed_id, ed_lname, ed_fname, ed_pos, phone, address, city, state, zip}) \\
\text{titleditors(} & \text{ed_id, title_id, ed_ord}) \\
\text{titles(} & \text{title_id, title, type, pub_id, price, advance, ytd_sales, contract, notes, pubdate}) \\
\text{publishers(} & \text{pub_id, pub_name, address, city, state})
\end{align*}

2. THIS QUESTION IS FOR DATALOG

a. Find the first name of all people who have been either editors or authors

\begin{align*}
Q2a(fn):- & \text{authors(},_,fn,_,_,_,_,_,_.) \\
Q2a(fn):- & \text{editors(},_,fn,_,_,_,_,_,_,_,_,_.)
\end{align*}

Common errors:

- \text{Trying to join instead of taking union}
- \text{Forgetting to make the heads of the queries the same name}
- \text{Note: not an error if you assumed that people needed to be in the titleauthor table and titleditor table to be an actual author or editor.}

b. Find the titles of all books by an author with the last name ‘Smith’ where the author’s zip is greater than 15232

\begin{align*}
Q2b(title) :- & \text{authors(aid, ‘Smith’, _, _, _, _, _, _, zip), titleauthors(aid, tid, _, _),} \\
& \text{titles(tid, title, _, _, _, _, _, _, _, _, _, _), _ > 15232}
\end{align*}

Common errors:

- \text{Putting lname = ‘Smith’ in a separate clause at the end}
- \text{Forgetting to use the titles table to get the title rather than just the titleID}