EXAMPLE QUERIES FROM NOV 19 LECTURE

— movie titles rented by each costumer, names concat’ed together

SELECT c.accountid, c.firstname, c.lastname, GROUP_CONCAT(m.title)
FROM customer c
JOIN previousrental p ON c.accountid=p.accountid
JOIN video v USING(videoid)
JOIN movie m USING(movieid)
GROUP BY accountid

— total revenue from each movie, aggregation

SELECT m.title, SUM(p.cost)
FROM movie m
JOIN video v ON m.movieid=v.movieid
JOIN previousrental p ON v.videoid=p.videoid
GROUP BY m.title

— use of HAVING, condition on that total

SELECT m.title, SUM(p.cost) totrent
FROM movie m
JOIN video v ON m.movieid=v.movieid
JOIN previousrental p ON v.videoid=p.videoid
GROUP BY m.title
HAVING totrent>=5

— sum and count with outer join, use of IFNULL, on previous

SELECT m.title, IFNULL(SUM(p.cost),0), COUNT(p.cost)
FROM movie m
LEFT JOIN video v ON m.movieid=v.movieid
LEFT JOIN previousrental p ON v.videoid=p.videoid
GROUP BY m.title

— videos rented by each customer, past and current, with UNION

SELECT c.accountid, c.firstname, c.lastname, p.videoid
FROM customer c  
JOIN previousrental p ON c.accountid=p.accountid
UNION
SELECT c.accountid, c.firstname, c.lastname, r.videoid  
FROM customer c  
JOIN rental r ON c.accountid=r.accountid

— above with just id’s, UNION ALL keeps duplicates

SELECT accountid, videoid  
FROM previousrental  
UNION ALL  
SELECT accountid, videoid  
FROM rental

— above query as subquery, in FROM clause, note renaming of subquery  
— list customers and all videos, past and current

SELECT c.accountid, c.firstname, c.lastname, ar.videoid  
FROM customer c JOIN  
(SELECT accountid, videoid  
FROM previousrental  
UNION ALL  
SELECT accountid, videoid  
FROM rental) AS ar  
ON c.accountid=ar.accountid

— as above, but COUNTS number of videos

SELECT c.accountid, c.firstname, c.lastname, COUNT(ar.videoid)  
FROM customer c JOIN  
(SELECT accountid, videoid  
FROM previousrental  
UNION ALL  
SELECT accountid, videoid  
FROM rental  
UNION ALL  
SELECT accountid, videoid  
FROM reservation) AS ar  
ON c.accountid=ar.accountid
GROUP BY c.accountid, c.firstname, c.lastname
as above, include costs, OUTER join to include all customers
- note COUNT of attribute gets number of non-nulls

```
SELECT c.accountid, c.firstname, c.lastname,
COUNT(ar.videoid) numrent,
IFNULL(SUM(ar.cost), 0) totspent
FROM customer c LEFT JOIN
(SELECT accountid, videoid, cost
FROM previousrental
UNION ALL
SELECT accountid, videoid, cost
FROM rental
UNION ALL
SELECT accountid, videoid, 0 as cost
FROM reservation) AS ar
ON c.accountid=ar.accountid
GROUP BY c.accountid, c.firstname, c.lastname
```