SQL Practice
Schema

borrower

Customer_name
loan_number

loan

loan_number
amount
branch_name

depositor

Customer_name
account_num

account

account_num
amount
branch_name

branch

branch_name
branch_city
assets
Find the names of branches that have greater assets than some branch located in Brooklyn

```
select distinct T.branch_name
from branch as T, branch as S
where T.assets > S.assets
and S.branch_city = 'Brooklyn'
```
String Operation (naïve text search)

- percent % or *: matches any substring
- underscore _ or ?: matches any character

Find the names of depositors whose name includes the substring “Mary”

- `select customer_name`
- `from depositor`
- `where customer_name like '%Mary%'`
String Operation (cont.)

- Any street name with exactly 5 characters
  - ... like ‘_ _ _ _ _’
- More string operation:
  - concatenation (using “||”)
  - finding string length, extracting substrings, etc.

- Full text search
  - Postgresql 8.3 also supports full text search
  - http://www.postgresql.org/docs/8.3/static/textsearch.html for more details
Nested Query

- Find branches that have greater assets than some branch located in Brooklyn

\[
\text{select } \text{branch\_name} \\
\text{from } \text{branch} \\
\text{where } \text{assets} > \text{some} \\
(\text{select } \text{assets} \\
\text{from } \text{branch} \\
\text{where } \text{branch\_city} = \text{‘Brooklyn’})
\]
Quantifier

☐ Some
- 5 < some \{0,5,6\}? True
- 5 < some \{0,5\}? False
- 5 = some \{0,5\}? True
- 5 \neq\ some \{0,5\}? True

☐ All
- 5 < all \{0,5,6\}? False
- 5 = all \{4,5\}? False
- 5 \neq\ all \{4,6\}? True
Quantifier (cont.)

- Find customers who have an account at all branches located in Brooklyn

```sql
select d.customer_name
From depositor d, account a
Where d.account_num = a.account_num AND
    a.branch_name = all
    ( select branch_name
        From branch
        where branch_city = 'Brooklyn'
    )
```

- Wrong!
correct solution

```sql
SELECT distinct S.customer_name
FROM depositor as S
WHERE not exists ( 
    (SELECT branch_name
     FROM branch
     WHERE branch_city = 'Brooklyn')
    EXCEPT
    (SELECT R.branch_name
     FROM depositor as T, account as R
     WHERE T.account_number = R.account_number AND
     S.customer_name = T.customer_name )
)
```

- `exists r` iff `r ≠ Ø`
- `not exists r` iff `r = Ø`
- `X except Y` means `X-Y`
More experience on SQL

☐ Ready ...

☐ Go~~~
1. which supplier should be selected to place an order for a given part in a given region?

**Select** ......

**From** part, supplier, partsupp, nation, region

**Where**

\[
p_{\text{partkey}} = ps_{\text{partkey}} \\
\text{AND} \quad s_{\text{suppkey}} = ps_{\text{suppkey}} \\
\text{AND} \quad p_{\text{size}} = [\text{SIZE}] \\
\text{AND} \quad p_{\text{type}} \text{ like } '%[\text{TYPE}]' \\
\text{AND} \quad s_{\text{nationkey}} = n_{\text{nationkey}} \\
\text{AND} \quad n_{\text{regionkey}} = r_{\text{regionkey}} \\
\text{AND} \quad r_{\text{name}} = '[\text{REGION}]' \\
\text{AND} \quad ps_{\text{supplycost}} = ( \\
\]
Select min(ps_supplycost)
From partsupp, supplier, nation, region
Where p_partkey = ps_partkey
AND s_suppkey = ps_suppkey
AND s_nationkey = n_nationkey
AND n_regionkey = r_regionkey
AND r_name = '[REGION]'
2. how well the order priority system is working?

Counts the number of orders ordered in a given quarter of a given year in which at least one line item was received by the customer later than its committed date

```sql
Select o_orderpriority,
    count(*) as order_count
from orders
```
Where o_orderdate >= date '[DATE]'
AND o_orderdate < date '[DATE]' + interval '3' month
AND exists (  
    select *  
    From lineitem  
    where l_orderkey = o_orderkey  
    AND l_commitdate < l_receiptdate  
  )
  group by o_orderpriority
order by o_orderpriority;
3. who might be having problems with the parts that are shipped to them?

```
Select c_custkey,
c_name,
sum(l_extendedprice * (1 - l_discount)) as revenue,
c_acctbal,
n_name,
c_address,
c_phone,
c_comment
from customer,
orders,
lineitem,
nation
```
Where c_custkey = o_custkey
    and l_orderkey = o_orderkey
    and l_returnflag = 'R'
    and c_nationkey = n_nationkey

group by c_custkey,
    c_name,
    c_acctbal,
    c_phone,
    n_name,
    c_address,
    c_comment

order by revenue desc;