1 Short Answer

a) What is a foreign key and what constraints does this key enforce (in SQL)?

   If an attribute is a foreign key, that means it references an attribute in another table. This attribute must be either NULL or a value in that other table’s column.

b) In SQL++, list two commands that are part of the Data Definition Language

   
   Create Database
   Create Type
   Create Dataset
c) Does the block nested loop join approach require indexes? Why or why not?

No. Each tuple in the primary table is compared against each tuple in the second. Because all cross-wise comparisons are made, no indexes are needed.

d) Suppose there is a table R(a,b) where B(R) = 1000, T(R) = 50000, V(R,a) = 300 and V(R,b) = 200. Values of R.b range from -20 to 80. What is the estimated number of disk accesses it would take to retrieve all records in R where 30 ≤ R.b ≤ 60? Why? Assume R.b has an unclustered index.

\[
\text{Selectivity factor} \quad \frac{60-30}{80-20} = \frac{30}{100}
\]

\[
\text{Estimated Tuples} = T(R) \cdot \frac{30}{100} = \frac{50,000 \times 30}{100} = 15,000
\]

Because the table is unclustered, each would require its own I/O.

Since there are only 1000 blocks, it would be faster to read all blocks w/pipelined selection.

Selection 2 1000
e) Will a Map/Reduce job take more time if a straggler is a Map task or a Reduce task? Explain.

Since all reduce tasks must wait for all Map tasks, a straggler in map can more seriously impact total runtime.

f) For which of the following does a semi-structured approach have the largest benefit over RDBMS? Many-to-one, many-to-many, one-to-many or one-to-one? Why?

One-to-many.

Semi-structured data allows joining on nested collections, whereas this is not allowed in INF.
g) In our semi-structured data, what is the difference between a dataverse and a dataset?

A dataset is a single relation containing objects of a type.

A dataverse containing all datasets in the reference data.

h) If a database has queries that are analytical, rather than transactional, should we prefer duplication or partitioning?

Duplication

⇒ harder to maintain throughout if there are many writes. Analytical is read-heavy.

⇒ distributing data means the DB can distribute query requests, increasing throughput.
i) Suppose we have the schema $R(a,b)$ $S(b,c)$. When are $R.a$ and $S.c$ NULL in a FULL OUTER JOIN of $R$ and $S$?

1) $R.b$ has no match $S.b$ and $R.a$ is NULL

2) $S.b$ has no match $R.b$ and $S.c$ is NULL

3) There is a match $R.b = S.b$ but $R.a$ and $S.c$ are null.

j) In NoSQL (semi-structured) data bases, what is heterogeneity and why does the problem arise?

Heterogeneity is when an attribute’s type varies from object to object.

This is possible because semi-structured data does not enforce types as strongly as SQL.
k) Why would a DBMS prefer uncorrelated subqueries?

If a subquery is uncorrelated, it only needs to be run once.

Correlated subqueries must be re-run for each matching value.
This results in longer runtime.

1) Give an example of a safe, non-linear recursive datalog rule.

\[
\text{Path}(x, y) : - \text{Edge}(x, y) \\
\text{Path}(x, y) : - \text{Path}(x, z), \text{Path}(z, y)
\]
m) We have the schema $R(a, b), S(x, y)$. Given the following query, draw a parallel query plan that reduces the amount of data sent over the interconnection network. Assume all attributes are INTs and $B(R) = B(S)$.

```sql
SELECT R.a
FROM R, S
WHERE R.b = S.x AND
    S.y = 100
```

R has been hash partitioned on $R.b$ and $S$ is block partitioned

Since $R$ is hash partitioned on $R.b$, the data in $R$ does not need to be reshuffled.
2 SQL

For this question, use the following schema:

Company(cid, country, name)
Product(pid, name, description)
Store(sid, cid, location)
Sells(sid, pid, price)

a) Write a SQL query to find the names of all companies who sell only products that cost more than $50. You may assume price is an INT.

```
SELECT DISTINCT C.name
FROM Company as C, Store as St
WHERE 50 > ALL (SELECT S.price
                   FROM Sells as S, Store as St, Company as C2
                   WHERE S.sid = St.sid
                       AND St.cid = C2.cid
                       AND C2.cid = C.cid)
```
b) Draw two logical plans that accomplish this goal.

![Diagram of logical plan]

we did not cover RA for universal quantifiers, it will not be on the midterm.

---

c) Name four pieces of information you would need to find the cheapest physical plan, in terms of disk accesses.

`B(Company) B(Store) B(Sells)`
3 Datalog

Suppose we have the following predicates in a datalog database:

Person(pid, name)
Parent(parentid, childid)

a) Are these predicates extensional or intensional?

Extensional

b) Define a predicate SC(x, y) which is true if x and y are second-cousins, i.e. they share a great-grand parent. Use intermediate predicates as necessary. Assume x and y are not necessarily all present names and that x and y are IDs.

\[
SC(x, y) : \rightarrow \text{Person}(x, a), \text{Person}(y, b), \text{Parent}(i, x), \text{Parent}(j, x), \text{Parent}(k, y), \text{Parent}(l, y), \text{Parent}(m, l), \text{Parent}(n, l) \\
\text{Common great-grand parent}
\]

Alternate, non stratified

\[
\text{Siblings}(x, y) : \rightarrow \text{Parent}(a, x), \text{Parent}(a, y)
\]

\[
\text{Cousins}(x, y) : \rightarrow \text{Parent}(a, x), \text{Parent}(b, y), \text{Siblings}(a, b)
\]

\[
SC(x, y) : \rightarrow \text{Parent}(a, x), \text{Parent}(b, y), \text{Cousins}(a, b)
\]
4 Cost Estimation

Give the cost for the following physical plans. Assume the following clustered indexes (R.A and S.C) and no unclustered indexes:

![Diagram with tree-like structure showing cost calculations]

\[
B(R) + B(R)B(S) = \\
1000 + 1000 \times 4000 = \\
\text{Unlikely to fit in memory}
\]

Clustered Selection

Selectivity \( \frac{40 - 20}{100 - 0} = \frac{20}{100} \)

\[
\frac{1}{d} = \frac{20}{18} \cdot B(R) = 200
\]

Index Join

10,000 tuples from left, each join with

\( T(S) \) tuples from right = \( \frac{100,000}{30,000} = 2 \) tuples

\[ 10,000 \times 2 = 20,000 \]

Total = 201,200