

**Lecture Notes 11**  
**CSE 350 Spring 2026**

**Quiz 3/3/2026**

**Q1:** Given a B+-tree (for a key-value store) with fanout  $F$ , leaf capacity  $C$  and height  $h$ , what is the maximum number of key-value pairs in it?

fanout: max number of child links in an internal node leaf capacity: max number of key-value pairs in a leaf node height: the number of nodes from the tree root to any leaf node

answer:  $CF^{h-1}$ .

**Q2:** Given Q1, prove that the lower bound of tree height of a B+-tree with  $N$  items is  $\Omega(\log_F N/C)$  but not  $\omega(\log_F N/C)$ .

$$N \leq CF^{h-1} \Rightarrow h \geq \log_F(N/C) + 1.$$

Clearly  $\log_F(N/C) + 1 = \Omega(\log_F N/C)$ .

To see why  $\log_F(N/C)+1 \notin \omega(\log_F N/C)$ , we need to show the following is not true:  $\lim_{n \rightarrow +\infty} \frac{\log_F N/C}{\log_F(N/C)+1}$  either does not exist or is not zero. But this limit is actually 1, which contradicts.

**Q3:** Suppose each B+-tree node spans exactly one disk page. What is the lower bound of the number of I/Os in pages for looking up the value associated with a particular key  $k$  in a B+-tree with fanout  $F$ , leaf capacity  $C$  and  $N$  key-value pairs?

Clearly,  $h \geq \log_F(N/C) + 1$ .

**Q4:** Given the following table schema:

Transaction(txid, custid, storeid, date, amt)

Answer the following queries using both standard relational algebra (with set semantics) and standard SQL:

1. Find all unique IDs of the customers who have visited both store 1000 and store 1001 in February 2026.

$$T_1, T_2 \leftarrow \sigma_{date \geq '2026-02-01' \wedge date \leq '2026-02-28'} Transaction$$
$$\pi_{T_1.custid}(\sigma_{storeid=1000} T_1 \bowtie_{T_1.custid=T_2.custid} \sigma_{storeid=1001} T_2)$$

```
SELECT DISTINCT T1.custid
FROM Transaction T1, Transaction T2
WHERE T1.custid = T2.custid
  AND T1.date >= '2026-02-01' AND T1.date <= '2026-02-28'
  AND T2.date >= '2026-02-01' AND T2.date <= '2026-02-28'
  AND T1.storeid = 1000
  AND T2.storeid = 1001
```

Alternative answer:

$$(\pi_{custid} \sigma_{date \geq '2026-02-01' \wedge date \leq '2026-02-28' \wedge storeid=1000} Transaction) \cap (\pi_{custid} \sigma_{date \geq '2026-02-01' \wedge date \leq '2026-02-28' \wedge storeid=1001} Transaction)$$

```
SELECT custid
FROM Transaction
WHERE date >= '2026-02-01' AND date <= '2026-02-28'
  AND storeid = 1000
INTERSECT
SELECT custid
FROM Transaction
WHERE date >= '2026-02-01' AND date <= '2026-02-28'
  AND storeid = 1001
```

2. Find all unique IDs of the customers who have visited store 1000 but not store 1001 in February 2026.

$$(\pi_{custid} \sigma_{date \geq '2026-02-01' \wedge date \leq '2026-02-28' \wedge storeid=1000} Transaction) - (\pi_{custid} \sigma_{date \geq '2026-02-01' \wedge date \leq '2026-02-28' \wedge storeid=1001} Transaction)$$

```
SELECT custid
FROM Transaction
WHERE date >= '2026-02-01' AND date <= '2026-02-28'
  AND storeid = 1000
EXCEPT
SELECT custid
FROM Transaction
WHERE date >= '2026-02-01' AND date <= '2026-02-28'
  AND storeid = 1001
```

Alternative SQL:

```
SELECT custid
FROM Transaction T1
WHERE date >= '2026-02-01' AND date <= '2026-02-28'
      AND storeid = 1000
      AND NOT EXIST(
SELECT *
FROM Transaction T2
WHERE date >= '2026-02-01' AND date <= '2026-02-28'
      AND storeid = 1001 AND T2.custid = T1.custid)
```

```
SELECT custid
FROM Transaction
WHERE date >= '2026-02-01' AND date <= '2026-02-28'
      AND storeid = 1000
      AND 0 = (
SELECT COUNT(*)
FROM Transaction
WHERE date >= '2026-02-01' AND date <= '2026-02-28'
      AND storeid = 1001 AND T2.custid = T1.custid)
```

```
SELECT custid
FROM Transaction
WHERE date >= '2026-02-01' AND date <= '2026-02-28'
      AND storeid = 1000
      AND 1001 NOT IN (
SELECT storeid
FROM Transaction
WHERE date >= '2026-02-01' AND date <= '2026-02-28'
      AND T2.custid = T1.custid)
```