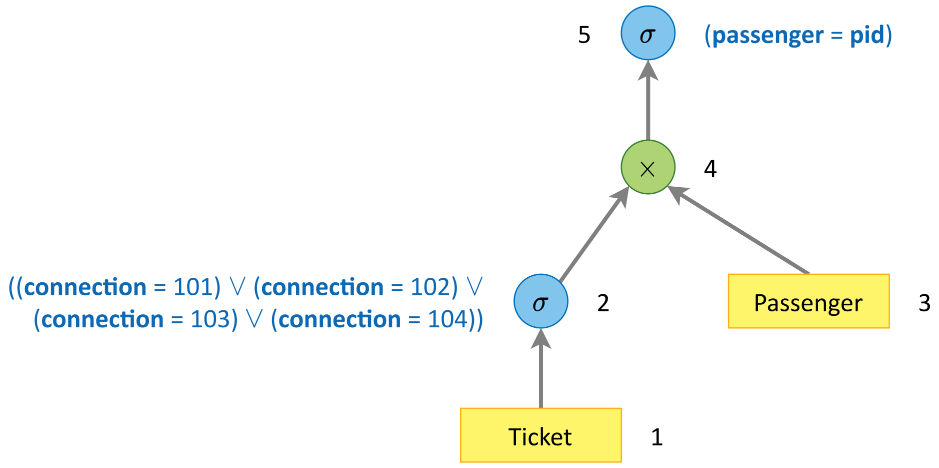
B0B36DBS: **Database Systems** | Class 9: **Query Evaluation**

Exercise 1


5
σ
(passenger = pid)
×
4
((connection = 101) ∨ (connection = 102) ∨
(connection = 103) ∨ (connection = 104))
σ
2
Passenger
3
Ticket
1

Evaluation plan

1. **Table** `Ticket`
   - **Heap file**
   - $r_1 = r_T = 900000$ records, $b_1 = b_T = 40$, $p_1 = r_1/b_1 = 900000/40 = 22500$ pages
2. **Selection** ((`connection` = 101) ∨ (`connection` = 102) ∨ (`connection` = 103) ∨ (`connection` = 104))
   - Reduction factor for four selected connections $f_{connections} = 4/V_{T.connection} = 4/20000 = 0.0002$
   - $r_2 = r_1 \cdot f_{connections} = 900000 \cdot 0.0002 = 180$ records
   - $b_2 = b_1 = 40$
   - $p_2 = \lceil r_2/b_2 \rceil = 180/40 = 5$ pages
   - Read cost (**sequential scan**) $c_2^r = p_1 = 22500$
   - Write cost $c_2^w = p_2 = 5$
3. **Table** `Passenger`
   - **Sorted file** using `pid`
   - $r_3 = r_P = 30000$ records, $b_3 = b_P = 15$, $p_3 = r_3/b_3 = 30000/15 = 2000$ pages
4. **Cross join** ×
   - $r_4 = r_2 \cdot r_3 = 180 \cdot 30000 = 5400000$ records
   - $b_4 = \lfloor (b_2 \cdot b_3)/(b_2 + b_3) \rfloor = \lfloor (40 \cdot 15)/(40 + 15) \rfloor = \lfloor 600/55 \rfloor = 10$
   - $p_4 = r_4/b_4 = 5400000/10 = 540000$ pages
   - **Nested loops** with zig-zag improvement
   - Memory configuration: $M_4 = [M_4^2 = 3$ as input buffer for sequential read of smaller table $T_2] + [M_4^3 = 1$ as input buffer for zig-zag reading of bigger table $T_3] + [1$ for join output buffer$] = 5$ pages
   - Read cost (join execution) $c_4^r = p_2 + \lceil p_2/M_4^2 \rceil \cdot (p_3 - M_4^3) + M_4^3 = 5 + \lceil 5/3 \rceil \cdot (2000-1) + 1 = 5 + 2 \cdot 1999 + 1 = 4004$
   - Write cost $c_4^w = p_4 = 540000$
5. **Selection** (`passenger` = `pid`)
   - Original **foreign key**: `Ticket` (`passenger`) ⊆ `Passenger` (`pid`)
   - Reduction factor $f_{theta} = 1/r_P = 1/30000$
   - I.e., only one appropriate passenger actually exists for a given ticket
   - Reduction factor for non-anonymous tickets $f_{registered} = 1 - h_{T.passenger.NULL} = 1 - 1/3 = 2/3$

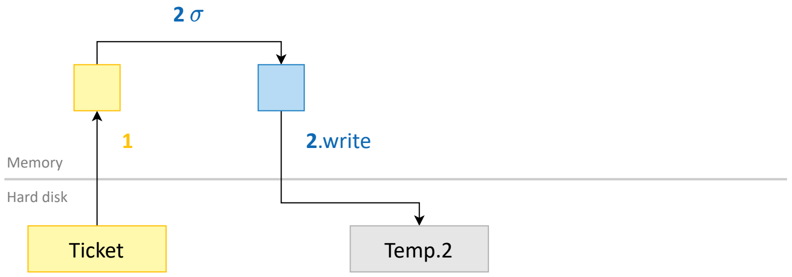
- $r_5 = r_4 \cdot f_{theta} \cdot f_{registered} = 5400000 \cdot 1/30000 \cdot 2/3 = 120$ records
- $b_5 = b_4 = 10$
- $p_5 = r_5/b_5 = 120/10 = 12$ pages
- Read cost $c_5^r = p_4 = 540000$
- Write cost $c_5^w = p_5 = 12$ (if not directly forwarded to the user)

Evaluation without pipelining

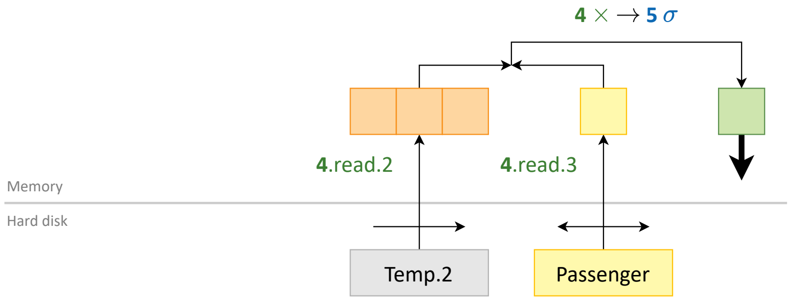
- Evaluation cost
  - $c = [c_2^r + c_2^w] + [c_4^r + c_4^w] + [c_5^r]$
  - $c = [p_1 + p_2] + [p_2 + \lceil p_2/M_4^2 \rceil \cdot (p_3 - M_4^3) + M_4^3 + p_4] + [p_4]$
  - $c = [22500 + 5] + [5 + \lceil 5/3 \rceil \cdot (2000 - 1) + 1 + 540000] + [540000]$
  - $c = [22505] + [544004] + [540000]$
  - $c = 1106509$

Evaluation with pipelining

- Memory configuration
  - Step 1: reading and filtering tickets

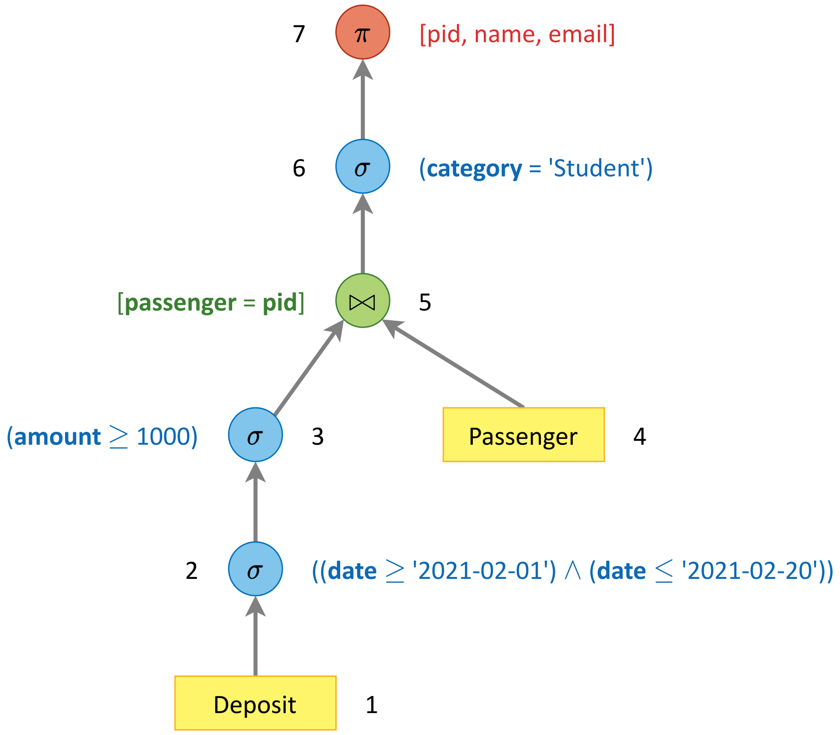


  - Step 2: reading passengers and joining them with filtered tickets



- Evaluation cost
  - $c = [c_2^r + c_2^w] + [c_4^r + \cancel{c_4^w}] + [\cancel{c_5^r}]$
  - $c = [p_1 + p_2] + [p_2 + \lceil p_2/M_4^2 \rceil \cdot (p_3 - M_4^3) + M_4^3]$
  - $c = [22500 + 5] + [5 + \lceil 5/3 \rceil \cdot (2000 - 1) + 1]$
  - $c = [22505] + [4004]$
  - $c = 26509$

Exercise 2



Evaluation plan

1. **Table** `Deposit`
   - **Sorted file** using `date` and `time`
   - $r_1 = r_D = 300000$ records, $b_1 = b_D = 60$, $p_1 = r_1/b_1 = 300000/60 = 5000$ pages
2. **Selection** ((`date` $\geq$ '2021-02-01') $\wedge$ (`date` $\leq$ '2021-02-20'))
   - Active domain for dates: $min_{D.date}$ ='2021-01-01' and $max_{D.date}$ ='2021-04-10'
   - I.e., $V_{C.date} \doteq (max_{D.date} - min_{D.date} + 1) = 100$ different dates
   - Reduction factor for permitted dates $f_{dates} =$ ('2021-02-20' $-$ '2021-02-01' $+ 1)$ / $V_{C.date} = 20/100 = 0.2$
   - $r_2 = r_1 \cdot f_{dates} = 300000 \cdot 0.2 = 60000$ records
   - $b_2 = b_1 = 60$
   - $p_2 = r_2/b_2 = 60000/60 = 1000$ pages
   - Read cost (**binary search**) $c_2^r = \log_2(p_1) + p_1 \cdot f_{dates} = \log_2(5000) + 5000 \cdot 0.2 = 13 + 1000 = 1013$
   - Write cost $c_2^w = p_2 = 1000$
3. **Selection** (`amount` $\geq$ 1000)
   - Reduction factor for amounts $f_{amounts} = h_{D.amount.[1000..)} = 0.05$
   - $r_3 = r_2 \cdot f_{amounts} = 60000 \cdot 0.05 = 3000$ records
   - $b_3 = b_2 = 60$
   - $p_3 = r_3/b_3 = 3000/60 = 50$ pages
   - Read cost $c_3^r = p_2 = 1000$
   - Write cost $c_3^w = p_3 = 50$
4. **Table** `Passenger`
   - **Sorted file** using `pid`
   - $r_4 = r_P = 30000$ records, $b_4 = b_P = 15$, $p_4 = r_4/b_4 = 30000/15 = 2000$ pages

5. **Theta join** [`passenger = pid`]

    - Original **foreign key**: `Deposit` (`passenger`) $\subseteq$ `Passenger` (`pid`)
    - Currently represented as T_3 (`passenger`) $\subseteq$ T_4 (`pid`)
    - $r_5 = r_3 = 3000$ records
    - $b_5 = (b_3 \cdot b_4)/(b_3 + b_4) = (60 \cdot 15)/(60 + 15) = 900/75 = 12$
    - $p_5 = r_5/b_5 = 3000/12 = 250$ pages
    - **Nested loops**, direct passenger look-up using **clustered index** over the `Passenger` table
    - Memory configuration: $M_5 = [1$ as input buffer for T_3 sequential read$] + [I_{P.pid} = 2$ for index nodes$] + [1$ as input buffer for T_4 retrieval$] + [1$ for join output buffer$] = 5$ pages
    - Root index node remains loaded in the memory all the time, leaf nodes are fetched iteratively on demand
    - Individual read costs (nested loops with index look-up) $c_5^{r.3} = p_3 = 50$ and $c_5^{r.4} = r_3 \cdot (I_{P.pid} - 1 + 1) + 1 = 3000 \cdot (2 - 1 + 1) + 1 = 6001$
    - Overall read cost $c_5^r = c_5^{r.3} + c_5^{r.4} = 50 + 6001 = 6051$
    - Write cost $c_5^w = p_5 = 250$

6. **Selection** (`category` = 'Student')

    - Reduction factor for student passengers $f_{category} = h_{P.category.Student} = 0.3$
    - $r_6 = r_5 \cdot f_{category} = 3000 \cdot 0.3 = 900$ records
    - $b_6 = b_5 = 12$
    - $p_6 = r_6/b_6 = 900/12 = 75$ pages
    - Read cost $c_6^r = p_5 = 250$
    - Write cost $c_6^w = p_6 = 75$

7. **Projection** [`pid`, `name`, `email`]

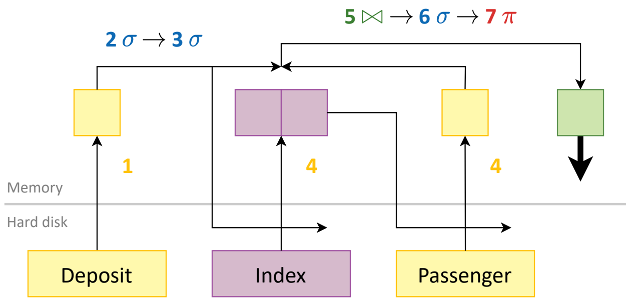
    - Resulting table contains a key for passengers (`pid`) and no deposit attributes
    - Each passenger is expected to have $n_{deposits.all} = r_D / V_{D.passenger} = 300000/30000 = 10$ deposit transactions during the whole period, and so just $n_{deposits} = n_{deposits.all} \cdot f_{dates} = 10 \cdot 0.2 = 2$ during the selected period
    - Reduction factor for removal of duplicates $f_{distinct} = 1/n_{deposits} = 1/2 = 0.5$
    - $r_7 = r_6 \cdot f_{distinct} = 900 \cdot 0.5 = 450$ records
    - $b_7 = 30$
    - $p_7 = r_7/b_7 = 450/30 = 15$ pages
    - Read cost $c_7^r = p_6 = 75$
    - Write cost $c_7^w = p_7 = 15$ (if not directly forwarded to the user)

Evaluation without pipelining

- Evaluation cost
    - $c = [c_2^r + c_2^w] + [c_3^r + c_3^w] + [c_5^r + c_5^w] + [c_6^r + c_6^w] + [c_7^r]$

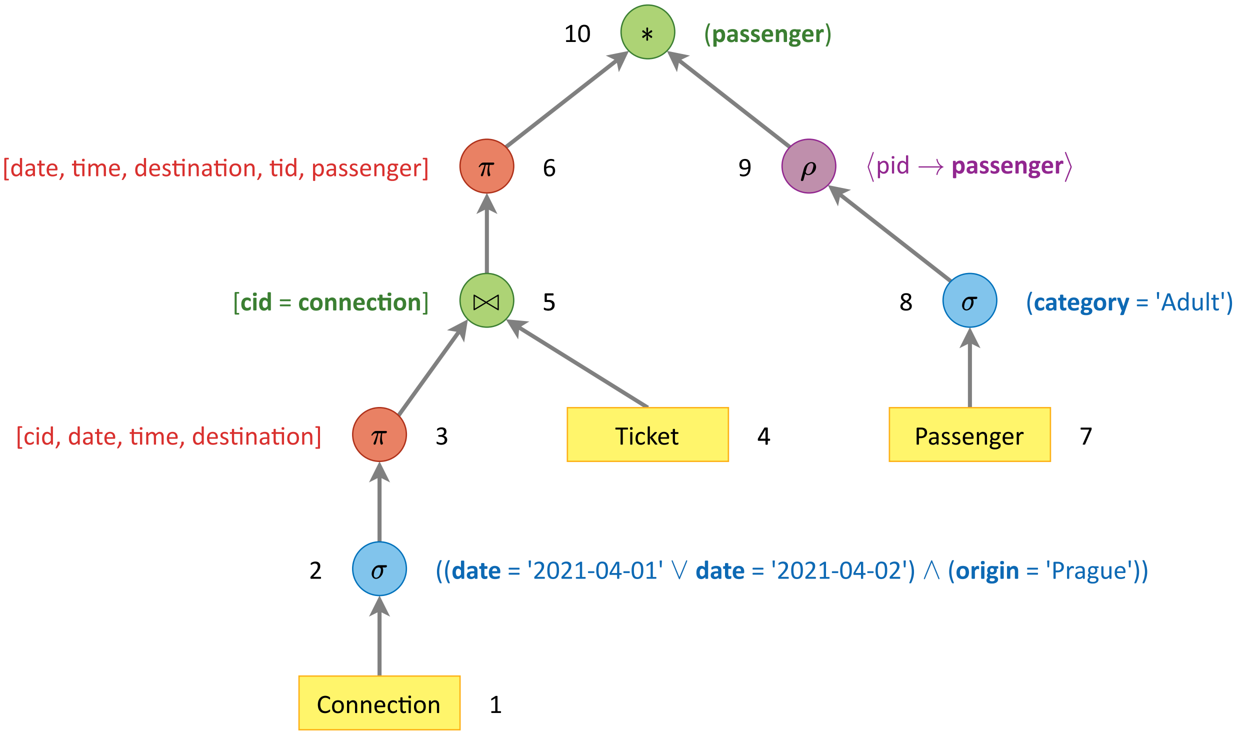
Evaluation with pipelining

- Memory configuration
    - Step 1: joining deposits and passengers



- Evaluation cost
  - $c = [c_2^r + \xcancel{c_2^w}] + [\xcancel{c_3^r} + \xcancel{c_3^w}] + [\xcancel{c_5^{r.3}} + c_5^{r.4} + \xcancel{c_5^w}] + [\xcancel{c_6^r} + \xcancel{c_6^w}] + [\xcancel{c_7^r}]$
  - $c = [\log_2(p_1) + p_1 \cdot f_{dates}] + [r_3 \cdot (I_{P.pid} - 1 + 1) + 1]$
  - $c = [\log_2(5000) + 5000 \cdot 0.2] + [3000 \cdot (2 - 1 + 1) + 1]$
  - $c = [1013] + [6001]$
  - $c = 7014$

Exercise 3


10
∗
(passenger)
[date, time, destination, tid, passenger]
π
6
9
ρ
⟨pid → passenger⟩
[cid = connection]
⋈
5
8
σ
(category = 'Adult')
[cid, date, time, destination]
π
3
Ticket
4
Passenger
7
2
σ
((date = '2021-04-01' ∨ date = '2021-04-02') ∧ (origin = 'Prague'))
Connection
1

Evaluation plan

1. **Table** `Connection`
   - **Hashed file** using `date`, $K_C = 80$ buckets
   - $r_1 = r_C = 20000$ records, $b_1 = b_C = 50$, $p_1 = r_1/b_1 = 20000/50 = 400$ pages
   - $C_1 = p_1/K_C = 5$ pages per bucket
2. **Selection** ((`date` = '2021-04-01' ∨ `date` = '2021-04-02') ∧ (`origin` = 'Prague'))
   - Active domain for dates: $min_{C.date}$ ='2021-01-01' and $max_{C.date}$ ='2021-04-10'
   - I.e., $V_{C.date} = 100$ different dates
   - Reduction factor for two permitted dates $f_{dates} = 2 \cdot 1/V_{C.date} = 2 \cdot 1/100 = 1/50$
   - Assumption that $s = 2$ buckets are needed to be scanned
   - Reduction factor for the city of origin $f_{origin} = h_{C.origin.Prague} = 1/2$
   - $r_2 = r_1 \cdot f_{dates} \cdot f_{origin} = 20000 \cdot 1/50 \cdot 1/2 = 200$ records
   - $b_2 = b_1 = 50$
   - $p_2 = r_2/b_2 = 200/50 = 4$ pages
   - Read cost (**bucket retrieval**) $c_2^r = 2 \cdot C_1 = 2 \cdot 5 = 10$
   - Write cost $c_2^w = p_2 = 4$

3. **Projection** [`cid`, `date`, `time`, `destination`]

   - $r_3 = r_2 = 200$ records, $b_3 = 60$, $p_3 = r_3/b_3 = 200/60 \doteq 4$ pages
   - Read cost $c_3^r = p_2 = 4$
   - Write cost $c_3^w = p_3 = 4$

4. **Table** `Ticket`

   - **Heap file**
   - $r_4 = r_T = 900000$ records, $b_4 = b_T = 40$, $p_4 = r_4/b_4 = 900000/40 = 22500$ pages

5. **Theta join** [`cid` = `connection`]

   - Original **foreign key**: `Ticket` (`connection`) $\subseteq$ `Connection` (`cid`)
   - Currently represented as T_4 (`connection`) $\subseteq T_3$ (`cid`)
   - Assumption of $n_5 = r_4/V_{T.connection} = r_4/r_1 = 900000/20000 = 45$ sold tickets per connection
   - $r_5 = r_3 \cdot n_5 = 200 \cdot 45 = 9000$ records
   - $b_5 = (b_3 \cdot b_4)/(b_3 + b_4) = (60 \cdot 40)/(60 + 40) = 2400/100 = 24$
   - $p_5 = r_5/b_5 = 9000/24 = 375$ pages
   - **Nested loops**, smaller left table T_3 entirely fits the available system memory
   - Memory configuration: $M_5 = [p_3 = 4$ for whole $T_3] + [1$ for T_4 input buffer$] + [1$ for join output buffer$] = 6$ pages
   - Individual read costs (nested loops execution) $c_5^{r.3} = p_3 = 4$ and $c_5^{r.4} = p_4 = 22500$
   - Overall read cost $c_5^r = c_5^{r.3} + c_5^{r.4} = p_3 + p_4 = 4 + 22500 = 22504$
   - Write cost $c_5^w = p_5 = 375$

6. **Projection** [`date`, `time`, `destination`, `tid`, `passenger`]

   - $r_6 = r_5 = 9000$ records, $b_6 = 30$, $p_6 = r_6/b_6 = 9000/30 = 300$ pages
   - Read cost $c_6^r = p_5 = 375$
   - Write cost $c_6^w = p_6 = 300$

7. **Table** `Passenger`

   - **Sorted file** using `pid`
   - $r_7 = r_P = 30000$ records, $b_7 = b_P = 15$, $p_7 = r_7/b_7 = 30000/15 = 2000$ pages

8. **Selection** (`category` = 'Adult')

   - Reduction factor for adult passengers $f_{category} = h_{P.category.Adult} = 0.5$
   - $r_8 = r_7 \cdot f_{category} = 30000 \cdot 0.5 = 15000$ records
   - $b_8 = b_7 = 15$
   - $p_8 = r_8/b_8 = 15000/15 = 1000$ pages
   - Note that the original sorting is maintained
   - Read cost (**sequential scan**) $c_8^r = p_7 = 2000$
   - Write cost $c_8^w = p_8 = 1000$

9. **Attribute renaming** $\langle$`pid` $\rightarrow$ `passenger`$\rangle$

   - $r_9 = r_8 = 15000$ records, $b_9 = b_8 = 15$, $p_9 = p_8 = 1000$ pages

10. **Natural join** (`passenger`)

    - Original **foreign key**: `Ticket` (`passenger`) $\subseteq$ `Passenger` (`pid`)
    - Currently represented as T_6 (`passenger`) $\subseteq T_9$ (`passenger`)
    - Reduction factor for non-anonymous tickets $f_{registered} = 1 - h_{T.passenger.NULL} = 1 - 1/3 = 2/3$
    - Reduction factor for adult passengers $f_{category} = h_{P.category.Adult} = 0.5$
    - $r_{10} = r_6 \cdot f_{registered} \cdot f_{category} = 9000 \cdot 2/3 \cdot 0.5 = 3000$ records
    - $b_{10} \doteq (b_6 \cdot b_9)/(b_6 + b_9) = (30 \cdot 15)/(30 + 15) = 450/45 = 10$
    - $p_{10} = r_{10}/b_{10} = 300$

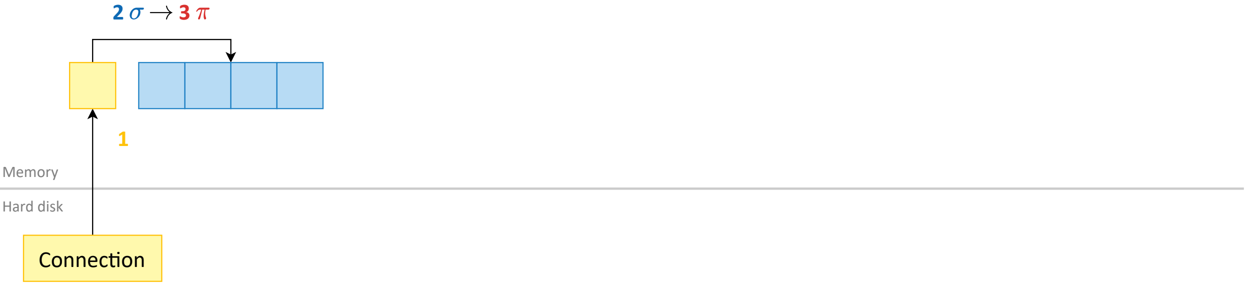
- **Sort-merge join** algorithm, left table T_6 needs to be sorted, right table T_9 is already sorted using $\mathtt{passenger}$
- **2-passes only**, integrated **priority queue**, extension allowing duplicates in one table (T_6)
- $M_6^{1.container} = \sqrt{p_6/2} = \sqrt{300/2} = \sqrt{150} \doteq 13$ pages are needed for priority queue container to ensure 2-passes
- Sorting phase (pass 1) is likely to produce ≈ 13 runs with length of $\approx 2 \cdot 13 = 26$ pages each
- Memory configuration (sorting phase for T_6): $M_6^1 = [M_6^{1.container} = 13$ for priority queue container$] + [1$ for T_6 input buffer$] + [1$ for sorting output buffer$] = 15$ pages
- Read cost $c_{10}^{r.sort.6} = p_6 = 300$
- Write cost $c_{10}^{w.sort.6} \doteq p_6 = 300$
- Memory configuration (joining phase): $M_6^2 = [M_6^{1.container} = 13$ as input buffer for individual runs of presorted $T_6] + [1$ as input buffer for the only run of already sorted $T_9] + [1$ for join output buffer$] = 15$ pages
- Individual read costs $c_{10}^{r.join.6} = c_{10}^{w.sort.6} \doteq p_6 = 300$ and $c_{10}^{r.join.9} = p_9 = 1000$
- Overall read cost $c_{10}^{r.join} = c_{10}^{r.join.6} + c_{10}^{r.join.9} \doteq p_6 + p_9 = 300 + 1000 = 1300$
- Write cost $c_{10}^{w.join} = p_{10} = 300$ (if not directly forwarded to the user)

Evaluation without pipelining

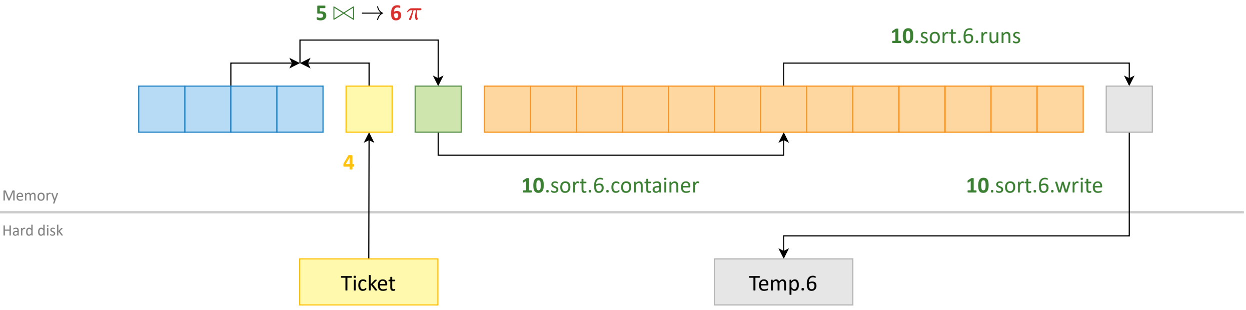
- Evaluation cost
  - $c = [c_2^r + c_2^w] + [c_3^r + c_3^w] + [c_5^r + c_5^w] + [c_6^r + c_6^w] + [c_8^r + c_8^w] + [c_{10}^{r.sort.6} + c_{10}^{w.sort.6} + c_{10}^{r.join}]$

Evaluation with pipelining

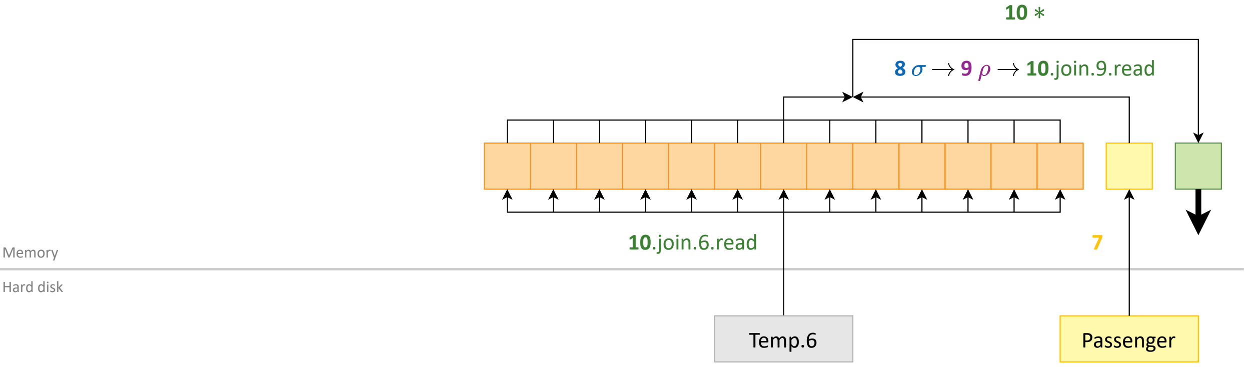
- Memory configuration
  - Step 1: reading, filtering and projecting connections



  - Step 2: reading tickets and joining them with connections, projecting extended tickets, and creating sorted runs



  - Step 3: joining extended tickets with passengers



- Evaluation cost
  - $c = [c_2^r + \cancel{c_2^w}] + [\cancel{c_3^r} + \cancel{c_3^w}] + [\cancel{c_5^{r.3}} + c_5^{r.4} + \cancel{c_5^w}] + [\cancel{c_6^r} + \cancel{c_6^w}] + [c_8^r + \cancel{c_8^w}] + [\cancel{c_{10}^{r.sort.6}} + c_{10}^{w.sort.6} + c_{10}^{r.join.6} + \cancel{c_{10}^{r.join.9}}]$
  - $c = [2 \cdot C_1] + [p_4] + [p_7] + [p_6 + p_6]$
  - $c = [2 \cdot 5] + [22500] + [2000] + [300 + 300]$
  - $c = [10] + [22500] + [2000] + [600]$
  - $c = 25110$