INNOPOLIS UNIVERSITY

Week 8 Lab

Query Optimization

[S25] Databases Course

Index creation

INNOPOLIS UNIVERSITY

CREATE INDEX index_name **ON** table_name [**USING** method](column_name [**ASC** | **DESC**] [**NULLS** {**FIRST** | **LAST** }]);

Explain result

	QUERY PLAN text
1	Seq Scan on actor a (cost=0.00..4.50 rows=23 width=25) (actual time=0.035..0.101 rows=23 loops=1)

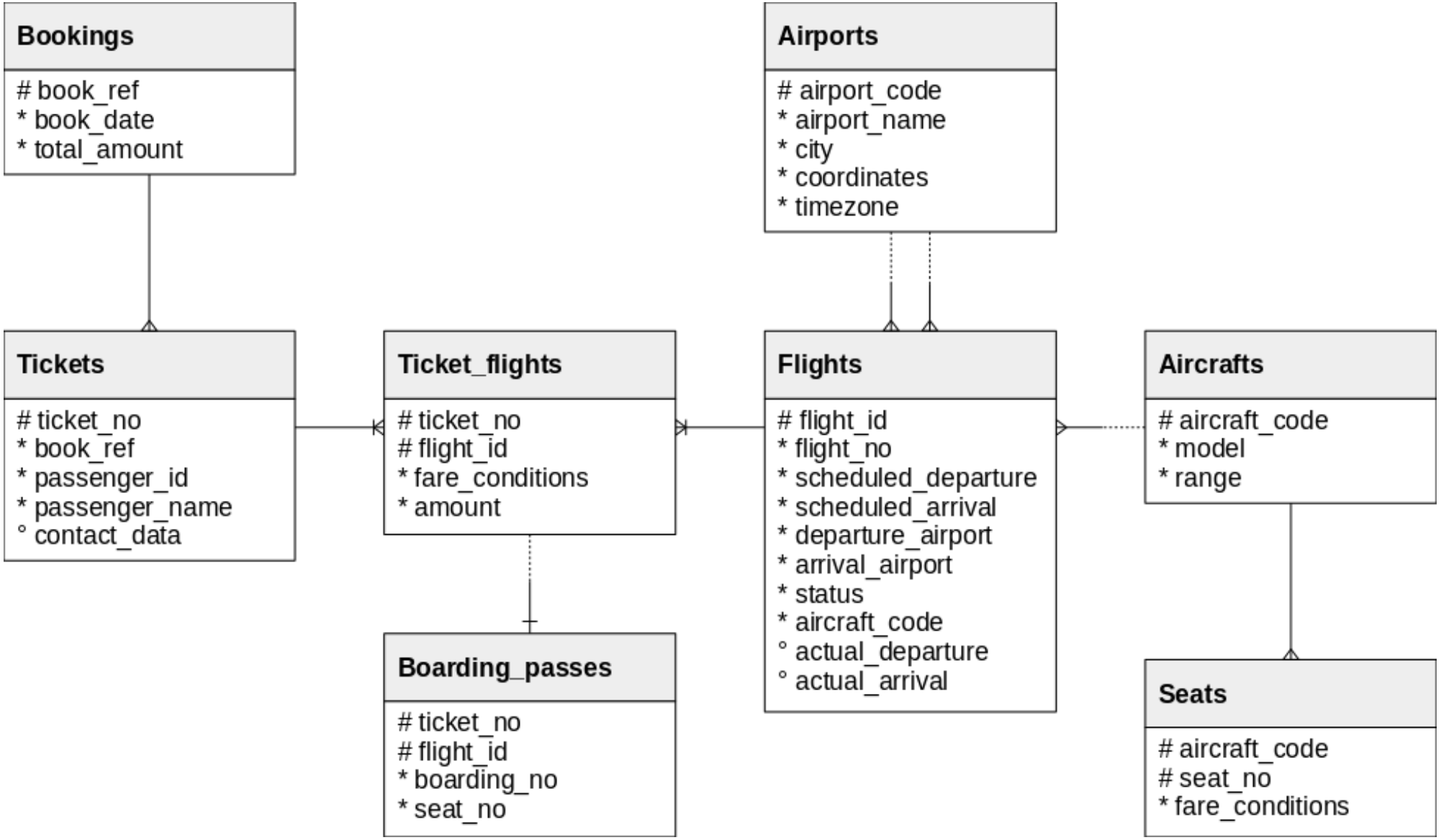
1 2 3 4 5

1. Types of scan nodes: sequential scans, index scans, and bitmap index scans (depends on the table access methods)
2. Estimated start-up cost (time expended before the output scan can start, e.g., time to do the sorting in a sort node)
3. **Estimated total cost (if all rows are retrieved, though they might not be; e.g., a query with a LIMIT clause will stop short of paying the total cost of the Limit plan node's input node)**
4. Estimated number of rows output by this plan node (again, only if executed to completion)
5. Estimated average width (in bytes) of rows output by this plan node

Lab Task

- Download the database from the following Link.
- Import the “demo-medium-en” database.
- Submission is individual.
- Submit your solution to Lab-08 directory.
- Submit each task in a sql file. Naming of the files should follow the convention: task1.sql, task2.sql, task3.sql

Schema Diagram



Task 1 - Index and Explain Analyze

- Find passenger_name, book_ref, ticket_no, book_date of bookings and tickets where *book_ref* starts with ‘B’ and *ticket_no* starts with '000543'.
- Print the analyze of the query execution time.
- Create the index `***book_ref_idx***` on *book_ref* from *bookings*.
- Execute the query again and print the analyze of the query execution time.
- Create the multi-column index `***book_ref_book_date_idx***` on *book_ref and book_date* from *bookings*.
- Execute the query again and print the analyze of the query execution time.
- Compare the results!!!

Task 2 - Index on Expressions

INNOPOLIS UNIVERSITY

- Find all the *ticket_flights* where *ticket_no* starts with ‘00054343’;
- Print the analyze of the query execution time.
- Create index `***ticket_no_index***` where *ticket_no* starts with ‘0005434’;
- Execute the query again and print the analyze of the query execution time.

Task 3 - Query

INNOPOLIS UNIVERSITY

- Optimize the following query.

Retrieve the ***flight_no*** of the *flights* for which tickets have been purchased by passengers whose names begin with the letter 'M'.

INNOPOLIS UNIVERSITY

Thank you for attention

See you next week

[S25] Databases Course