Investigate Database Recovery

Omi Johnson

Question 1

Data warehouses are often constructed using relational databases. Explain the use of fact tables and star schemas to construct a data warehouse in a relational database. Also comment on whether a transactional database can and should be used to OLAP.

Fact tables and star schemas are crucial elements to construct a data warehouse in a relational database. Data warehouses hold massive amounts of data and are designed to maximize query efficiency - therefore, they are stored in a non-normalized form so as to minimize the number of joins required during queries. A star schema eliminates complex joins and allows the entirety of the database to be accessed by just considering the foreign keys of a primary table.

In a star schema design, a data warehouse is built by having one primary table (the *fact table*) reference all the other dimension tables in the database. Each tuple within the fact table represents the "base interaction" of the database - for example, the purchasing of an item at the grocery store. For each dimension of the interaction, a dimension table can keep track of any additional attributes, allowing for multiple-dimension data (e.g. a purchased item has price, description, etc.). A *snowflake schema* is a version of a star schema that allows for additional tables not directly referenced by the fact table.

A transactional database should not be used for OLAP. OLAP is designed to query high volumes of multidimensional data at high speeds. While you can extract analyses from a transactional database, it is designed to optimize data manipulation and will perform poorly when asked to retrieve high volume queries.

Question 2

Explain the difference between a data warehouse, a data mart, and a data lake. Provide at least one example of their use from your experience or how you believe they might be used in practice. Find at least one video, article, or tutorial online that explains the differences and embed that into your notebook.

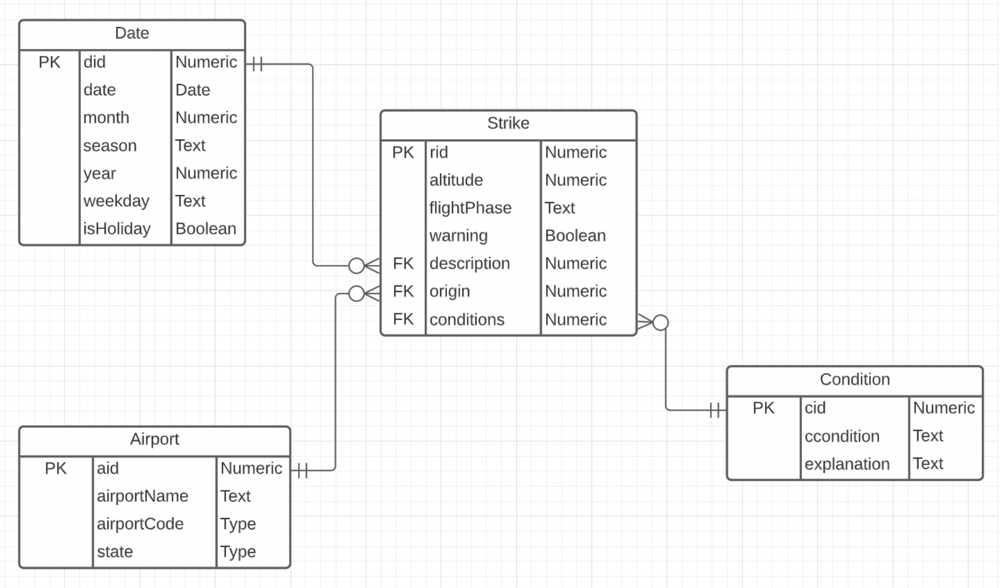
https://www.indeed.com/career-advice/career-development/data-lake-vs-data-warehouse-vs-data-mart

A **data warehouse** is a large collection of data from multiple input sources. It contains all relevant data for an organization in a single database to allow for easy analysis. The input data is transformed via ETL before being added. A **data mart** holds a summarized subset of the data from a larger database. It's best to store a subset of data within the database relevant to a particular subject to enable manageable querying. Finally, a **data lake** is a repository that holds raw data in its original format until usable. It's best to store unclassified data for later editing and has more flexibility than a data warehouse.

Question 3

After the general explanation of fact tables and star schemas, design an appropriate fact table for Practicum I's bird strike database. Of course, there are many fact tables one could build, so pick some analytics problem and design a fact table for that. Be sure to explain your approach and design reasons. Just design it (perhaps draw an ERD for it); you do not need to actually implement it or populate it with data (of course, you may do so if you wish in preparation for the next practicum).

```r
knitr::include_graphics("schema.png")
```



Date

PK	did	Numeric
	date	Date
	month	Numeric
	season	Text
	year	Numeric
	weekday	Text
	isHoliday	Boolean

Strike

PK	rid	Numeric
	altitude	Numeric
	flightPhase	Text
	warning	Boolean
FK	description	Numeric
FK	origin	Numeric
FK	conditions	Numeric

Condition

PK	cid	Numeric
	ccondition	Text
	explanation	Text

Airport

PK	aid	Numeric
	airportName	Text
	airportCode	Type
	state	Type

This fact table was built to examine how bird strikes are correlated to different contributing variables, with an emphasis on time of year and location. This data looks similar to the relational schema from practicum one, with "Strikes" acting as the fact table and Date, Airport, and Condition as dimension tables. Notable differences include the addition of the date table with a variety of derived attributes such as month, season, and weekday. Additionally, the Airport table has derived attributes based on the location of the airport. The additional dimension table and derived airport attributes allow for easier querying and sorting by date/location based attributes.