BrightLearn Data Analytics

Research Assignment 1

Section A: Database Fundamentals

1. What are main types of databases?

Relational Databases (RDBMS)
- Store data in tables with rows and columns (e.g, MySQL, PostgreSQL, oracle)

NOSQL Databases
- Store unstructured or semi-structured data (e.g, MongoDB, Cassandra)

Object-oriented Databases
- store data as objects (e.g, db4o)

Graph Databases
- Focus on relationships (e.g, Neo4j)

Time series Databases
- Optimised for time-based data (e.g, InfluxDB)

Cloud Databases
- Hosted on cloud platforms (e.g Snowflake, AWS RDS)

2. What is a Relational Database Management system (RDBMS)
- An RDMS is software that manages relational databases, where data is stored in in tables related by keys. It ensures data integrity,

supports SQL queries, and enforces relationships between tables
e.g, MySQL, SQL Server, Oracle, and PostgreSQL

3. What is a primary key and a foreign key in a database?
* Primary key
- Is a unique identifier for each record in a table (e.g, student_id in a students table)

Foreign key
- Is a field in one table that refers to the primary key in another table, creating a relationship

Example: Orders (customer_id) -> references -> Customers (customer_id)

4. What is a database normalization and why is it important?
- Normalization is the process of organizing data ~~into~~ to minimise redundancy and improve data integrity

- It splits large tables into smaller ones and defines relationships using keys

Benefits: - Less duplicate data, easier updates, improved consistency

Normal forms: 1NF, 2NF, 3NF, BCNF

5. What is a database schema?
- A schema is the logical structure of a database
- It defines tables, columns, relationship, constraints, and data types
- Its the blue print of how the data is organised
Example: sales_schema might include customers, orders, and products tables

6. Differentiate between structured, semi-structured and unstructured data?

Type	Description	Example
structured	Organized in rows and columns, easily queried with SQL, csv file	Relational DBs (e.g, MySQL)
Semi-structured	Has tags or structure but not fixed schema	JSON, XML, MongoDB
unstructured	No predefined format	Images, Videos, text, Emails, PDFs

7. What is the difference between a Fact Table and Dimension Table in a data warehouse?

Feature	Fact Table	Dimension Table
Purpose	stores measurable data (metrics)	stores descriptive attributes
Example	sales amount, quantity	Product, Customer, Date
Nature	Numaric, transcational	Textual, categorical
keys	Has foreign keys referencing dimensions	Has primary key

8. What is a data model, and why is it important in database design?

- A data model defines how data is structured, related, and stored

It is important because it:

- Ensures data consistency and clarity
- Simplifies database design and maintanance
- Helps developers and business users understand relanthanships

Types:
- conceptual
- Logical
- physical data models

9. Explain the difference between a database, a data warehouse, and a data lake

Feature	Database	Data Warehouse	Data Lake
purpose	Daily operations (OLTP)	Analytics and reporting (OLAP)	store raw, unprocessed data
Data Type	structured	structured (aggregated)	All types (structured, semi, unstructured)
Example	MySQL	Snowflake, Redshift	AWS S3, Azure Data Lake

10. What is a data mart, and how does it differ from data warehouse?

— A data mart is a subset of a data warehouse focused on a specific business area (e.g, sales, HR, marketing)

Feature	Data Warehouse	Data Mart
scope	Enterprise-wide	Department-specific
size	Large	Smaller
Data source	Multiple systems	Typically from the warehouse itself

SECTION B: SQL and Data Processing

11. What is a query language, and why is SQL the most commonly used?

— A query language allows users to interact with a database, to retrieve, insert, update, or delete data
— SQL (Structured Query Language) is the most widely used because it:
* Is standardized (ANSI SQL) across systems
* Works with most relational databases (MySQL, SQL Server, Oracle, Snowflake)
* Is declarative, you specify what you want, not how to get it

12. What are indexes in databases, and how do they improve performance?

— An index is a data structure (like a lookup table) that speeds up data retrieval
— It works like an index in a book, instead of scanning every page, you jump directly to where the info is
— Improves performance for SELECT queries but can slow down INSERT/UPDATE/DELETE since the index must be updated too

13. What are transactions in databases, and what are the ACID properties?

- A transaction is a single logical unit of work that must complete fully or not at all

ACID stands for:

Property	Description
A - Atomicity	All operations succeed or all fail
C - Consistency	The database remains in a valid state
I - Isolation	Transactions don't interfere with each other
D - Durability	once committed, data is permanent

14. What is a database engine, and how does it impact performance?

- A database engine is the core component that manages how data is stored, queried and processed

It determines:

- speed of reads/writes
- Query optimization
- Storage management

Different engines are optimized for different workloads:

Example: MySQL uses InnoDB, SQL server uses

MS SQL Engine, and Snowflake uses a cloud-native engine

15. What are views, stored procedures, and triggers in SQL?

Feature	Description	Example
View	A virtual table based on a query. Simplifies complex queries	CREATE VIEW high_salary AS SELECT * FROM employees WHERE salary > 50000;
Stored Procedure	A saved block of SQL code that performs a task (can include logic)	CREATE PROCEDURE increase_salary()
Trigger	Code that runs automatically in response to an event (INSERT/UPDATE/DELETE)	CREATE TRIGGER after insert

16. Differentiate between ETL (Extract, Transform, Load) and ELT (Extract, Load, Transform)

Step	ETL	ELT
Definition	Data is extracted, and then loaded into the target system	Data is extracted, loaded first, then transformed inside the target (e.g, Snowflake
Best for	Traditional data warehouses (on-premise	Cloud-based systems with high processing power
Example Tools	Informatica, Talend	Snowflake, BigQuery

11. Differentiate between batch processing and stream processing in data pipelines

Feature	Batch Processing	
Definition	Processes large data sets at once	processes data in real time as it arrives
use case	Payroll, daily reports	Fraud detection, IoT, stock prices
Example Tools	Apache Spark, AWS Batch	Apache kafka, Flink, spark streaming

18. Explain what a join is in SQL and list different types of join examples

— A join combines data from multiple tables based on a related column

Join Type	Description
INNER JOIN	Returns only matching records
LEFT JOIN	Returns all ~~not~~ from left table + maches from right
Right JOIN	Returns all from the right table + matches from the left tables
Full OUTER JOIN	Returns all records from both tables
CROSS JOIN	returns all combinations (cartesian product)

19. What is referential integrity, and why is it important in relational databases?

— Referential integrity ensures relationships between tables remain valid, a foreign key must always reference an existing primary key

20. How does data redundancy affect database performance and storage?

- Data redundancy = storing the same data in multiple places

Disadvantages:

- Increases storage cost
- Causes inconsistency (if one copy changes)
- Slows down updates and maintenance
- Controlled by normalization and relational design

SECTION C: Data Management and Analytics Concepts

21. How does cloud database management differ from on-premise databases?

Aspect	Cloud Database	On-Premise Database
Hosting	Managed on remote servers (AWS, Azure, GCP, Snowflake)	Installed on local company servers
Scalability	Auto-scales on demand	Requires manual hardware upgrades
Cost	Pay as you subscription	Up front hardware + maintenance cost
Maintenance	Managed by provider	Managed by internal IT Team
Access	Accessible from anywhere	Limited to local network

Example: Snowflake, and BigQuery are fully managed cloud-native data warehouses

22. What is a data governance, and why is it important?

- Data governance is the framework of policies and processes that define how data is managed, accessed, and protected

Importance:
- Ensures data accuracy, consistency, and security
- Defines roles and responsibilities
- Supports regulatory compliance (e.g., GDPR)
- Builds trust in enterprise data

23. What is data integrity, and how can it be maintained?

- Data integrity means the data remains accurate, consistent, and reliable throughout its lifecycle

Maintained by:
- Using primary and foreign keys to maintain valid relationships
- Applying constraints (NOT NULL, UNIQUE)
- Enforcing referential integrity
- Using transactions (ACID) to prevent partial updates

Example: A student's ID should always uniquely identify one student, enforced with a primary key

24. What is data quality, and why is it critical for analytics?

- Data quality measures how complete, accurate, timely, and consistent data is

Why it matters:
- Poor data → wrong insights and bad business decisions
- High quality data → trustworthy analytics and reporting

Dimensions of data quality:
- Accuracy
- Completeness
- Consistency
- Timeliness
- Validity

25. Explain the role of a Data Analyst in managing and analyzing database information

* A Data Analyst
- Extracts and cleans data using SQL, Python, Excel and so on
- Performs statistical analysis and reporting
- Builds dashboards (Power BI, Tableau)
- Identifies trends and insights for decision making
- Works with DBAs and engineers to ensure data is structured and reliable

Example: Writing SQL queries to calculate monthly sales trends

26. What are the key responsibilities of a Database Administrator (DBA)?

- A DBA ensures databases are secure, efficient, and available

Main duties:
- Install and configure DB systems
- Manage backups and recovery
- Tune performance (indexes, query optimization)
- Control user access and permissions
- Monitor system health and logs

Example: Restoring a corrupted database backup after a failure

27. What are the main steps involved in designing data pipeline?

- A data pipeline moves data from sources → destination (warehouse or analytics tool)

Main steps:
- Extract → get data from sources (APIs, databases)
- Transform → clean, validate, and format it
- Load → store it in a target system (ETL / ELT)
- Orchestrate → schedule and automate workflows (Airflow, dbt)
- Monitor → Ensure reliability and performance

Example: ETL pipeline moving CRM data → Snowflake → Power BI

28. What are some common challenges in managing large scale databases?

Challenges:
- Performance issues (slow queries, heavy joins)
- Scalability (handling millions of records)
- Storage costs
- Data security and privacy
- Backup and disaster recovery
- Integration with multiple systems
- Data quality and consistency

29. What are some popular database platforms and key cases?

Platform	Type	Key use case
MySQL	Relational	Web apps, small–medium systems
PostgreSQL	Relational + advanced SQL	Complex analytical queries, open source
Oracle DB	Enterprise relational	Mission-critical systems
SQL Server	Relational	Enterprise Windows environments
Snowflake	cloud data warehouse	Scalable analytics and BI
MongoDB	NO SQL	JSON-based, unstructured data
BigQuery / Redshift	cloud data warehouses	Big data analytics

30. What are the main data storage formats used in analytics?

Format	Type	description	use Case
CSV	Text	Simple comma-separated format	Data exchange, spreadsheets
JSON	semi-structured	key-value pairs, nested	web APIs, NO SQL
Parquet	columnar binary	Compressed, optimized for analytics	Big data (spark, snowflake)
Avro	row-based binary	schema-based compact	streaming data (kafka)
ORC	Columnar binary	Efficient compression	Hadoop, Hive