

BrightLight Data Analytics - Research Assignment 1

Solutions

Student Name

October 2025

Section A: Database Fundamentals

1. Main Types of Databases

- **Relational Databases (SQL):** Store data in tables with rows and columns (e.g., MySQL, PostgreSQL)
- **NoSQL Databases:**
 - Document databases (MongoDB)
 - Key-value stores (Redis)
 - Column-family stores (Cassandra)
 - Graph databases (Neo4j)
- **In-memory Databases:** Store data in RAM for fast access (Redis)
- **Time-series Databases:** Optimized for time-stamped data (InfluxDB)
- **NewSQL Databases:** Combine SQL reliability with NoSQL scalability (Google Spanner)

2. Relational Database Management System (RDBMS)

A software system that manages relational databases using SQL. It organizes data into tables with relationships and maintains data integrity through constraints and ACID properties.

3. Primary Key vs Foreign Key

- **Primary Key:** Unique identifier for each record in a table (cannot be NULL)
- **Foreign Key:** Field in one table that references the primary key in another table, establishing relationships

4. Database Normalization

The process of organizing data to reduce redundancy and improve integrity through normal forms (1NF, 2NF, 3NF).

Importance: Prevents data anomalies, reduces storage, ensures consistency

5. Database Schema

The logical structure of a database including tables, columns, relationships, and constraints. It defines how data is organized and how relationships are enforced.

6. Data Types Differentiation

- **Structured:** Organized in predefined format (SQL tables)
- **Semi-structured:** Flexible schema with some structure (JSON, XML)
- **Unstructured:** No predefined format (images, videos, documents)

7. Fact Table vs Dimension Table

- **Fact Table:** Contains quantitative measurements (facts) and foreign keys to dimensions
- **Dimension Table:** Contains descriptive attributes that provide context to facts

8. Data Model

A conceptual representation of data structures and relationships.

Importance: Ensures data integrity, guides database design, facilitates communication

9. Database vs Data Warehouse vs Data Lake

- **Database:** For transactional processing (OLTP)
- **Data Warehouse:** For analytical processing (OLAP), structured, historical data
- **Data Lake:** Stores raw data in native format, supports all data types

10. Data Mart

A subset of a data warehouse focused on a specific business line.

Difference: More specific, smaller scope, faster implementation than data warehouse

Section B: SQL and Data Processing

11. Query Language & SQL

A language for retrieving and manipulating data in databases.

SQL dominance: Standardized, powerful, widely supported, handles complex relationships

12. Indexes

Data structures that improve data retrieval speed.

Performance improvement: Reduces full table scans, faster searching and sorting

13. Transactions & ACID Properties

- **Transactions:** Sequence of database operations treated as single unit
- **ACID:**
 - Atomicity: All or nothing
 - Consistency: Valid state transitions
 - Isolation: Concurrent transactions don't interfere
 - Durability: Committed changes persist

14. Database Engine

The core component that stores, processes, and secures data.

Performance impact: Determines storage efficiency, query processing speed, concurrency handling

15. Views, Stored Procedures, Triggers

- **Views:** Virtual tables based on query results
- **Stored Procedures:** Precompiled SQL code blocks
- **Triggers:** Automated actions on database events

16. ETL vs ELT

- **ETL:** Extract, Transform, Load (transform before loading)
- **ELT:** Extract, Load, Transform (transform after loading, uses target system's power)

17. Batch vs Stream Processing

- **Batch Processing:** Processes data in large chunks at scheduled intervals
- **Stream Processing:** Processes data in real-time as it arrives

18. SQL Joins

- **INNER JOIN:** Returns matching records from both tables
- **LEFT JOIN:** All records from left table + matches from right
- **RIGHT JOIN:** All records from right table + matches from left
- **FULL OUTER JOIN:** All records when there's a match in either table
- **CROSS JOIN:** Cartesian product of both tables

19. Referential Integrity

Ensures relationships between tables remain consistent.

Importance: Prevents orphan records, maintains data consistency, enforces business rules

20. Data Redundancy Impact

- **Negative effects:** Wasted storage, update anomalies, inconsistency, increased backup size
- **Positive effects:** Sometimes improves read performance

Section C: Data Management and Analytics Concepts

21. Cloud vs On-premise Databases

- **Cloud:** Scalable, pay-as-you-go, managed services, automatic updates
- **On-premise:** Full control, higher security responsibility, capital expenditure

22. Data Governance

The overall management of data availability, usability, integrity, and security.

Importance: Ensures data quality, compliance, security, and proper usage

23. Data Integrity

The accuracy and consistency of data over its lifecycle.

Maintenance: Constraints, validation rules, proper database design, access controls

24. Data Quality

The measure of data's fitness for its intended use.

Critical for analytics: Poor quality leads to incorrect insights, bad decisions, wasted resources

25. Data Analyst Role

- Interpret data patterns and trends
- Create reports and visualizations
- Ensure data quality and accuracy
- Provide actionable insights to stakeholders
- Collaborate with database teams on requirements

26. Database Administrator Responsibilities

- Database installation and configuration
- Backup and recovery
- Performance tuning and optimization
- Security management
- Capacity planning
- User access management

27. Data Pipeline Design Steps

1. Requirements gathering
2. Source identification
3. Data extraction
4. Data transformation
5. Data loading
6. Quality validation
7. Monitoring and maintenance

28. Large-scale Database Challenges

- Performance optimization
- Scalability management
- Data security and privacy
- Backup and recovery complexity
- Cost management
- Data consistency across distributed systems

29. Popular Database Platforms

- **MySQL:** Web applications, open-source projects
- **Snowflake:** Cloud data warehousing, analytics
- **PostgreSQL:** Complex applications, geospatial data
- **Oracle:** Enterprise applications, large-scale systems
- **MongoDB:** Document storage, flexible schemas
- **Redis:** Caching, real-time applications

30. Data Storage Formats

- **CSV:** Simple, human-readable, poor for complex data
- **Parquet:** Columnar storage, efficient for analytics
- **JSON:** Flexible, good for semi-structured data
- **Avro:** Binary format, schema evolution, efficient storage
- **ORC:** Optimized columnar format for Hive