Window Functions:

Window functions perform calculations across a set of rows related to the current row.

An example is calculating a moving average or ranking rows based on certain criteria within a partition.

Common Table Expressions (CTEs):

CTEs are temporary result sets defined within the execution of a SELECT, INSERT, UPDATE, DELETE, or CREATE VIEW statement. They are similar to subqueries but offer better readability and reusability.

Indexes:

Indexes are data structures that improve the speed of data retrieval operations on a database table at the cost of additional space and slower writes. Common types include clustered, non-clustered, unique, and composite indexes.

Transactions:

A database transaction is a single unit of work performed within a database management system against a database. Transactions must satisfy the properties of ACID: Atomicity, Consistency, Isolation, and Durability.

Optimization Techniques:

Optimization techniques include using indexes, rewriting queries to minimize resource consumption, denormalization for performance gains, and analyzing query execution plans.

Database Design:

Database design involves creating a logical and physical schema, normalizing data to reduce redundancy, denormalizing for performance when necessary, and considering factors like data integrity, scalability, and performance.

Stored Procedures vs. Functions:

Stored procedures are sets of SQL statements that can be executed as a single unit, while functions return a single value based on input parameters. Stored procedures can perform DML operations, while functions are primarily used for computations.

Views:

Views are virtual tables generated from the result of a SELECT query. They provide a way to simplify complex queries, restrict access to specific columns, and encapsulate complex logic.

Database Triggers:

Triggers are special stored procedures that are automatically executed or fired when certain events occur in the database. They can be used to enforce business rules, maintain data integrity, or log changes.

Subqueries vs. Joins:

Subqueries are queries nested within another query, while joins combine columns from two or more tables based on related columns. Subqueries are typically used for one-time operations, while joins are used for joining tables in a more structured manner.

SQL Injection:

SQL injection is a code injection technique that exploits vulnerabilities in an application's software by injecting SQL code into data inputs. It can be prevented by using parameterized queries, input validation, and proper access controls.

Temporal Data:

Temporal data management involves handling data that changes over time. Techniques include using temporal tables, effective dating (storing start and end dates for each record), or system-versioned tables (storing historical versions of records).

Database Joins:

Joins are used to combine rows from two or more tables based on a related column between them. Common types include INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL JOIN.

Recursive Queries:

Recursive queries are SQL queries that reference themselves. They are commonly used to traverse hierarchical data structures, such as organizational charts or tree structures.

Normalization:

Normalization is the process of organizing data in a database to reduce redundancy and dependency. It involves dividing large tables into smaller, related tables and defining relationships between them. Normalization forms include 1NF, 2NF, 3NF, BCNF, and 4NF.

Alteryx for data extraction from Salesforce

Capability to select, filter, formula and cleanse

Data Science:

1. \*\*Experience with data preprocessing and cleaning:\*\*

"**In my previous role, I've dealt with various types of messy data, including missing values, outliers, and inconsistent formats. I typically start by identifying and handling missing values, followed by outlier detection and removal using statistical methods or domain knowledge. I also perform data normalization or scaling to ensure features are on a similar scale**."

1. \*\*Approach to feature selection and engineering:\*\*

"**I approach feature selection by first understanding the problem domain and the business objectives. I then use techniques like correlation analysis, feature importance from tree-based models**, or dimensionality reduction methods like PCA. **Feature engineering involves creating new features from existing ones to improve model performance. For example, I might derive temporal features from timestamps or create interaction terms between variables**."

1. \*\*Difference between supervised and unsupervised learning:\*\*

"Supervised learning involves training a model **on labeled data**, where the algorithm learns **to predict the target variable**. Examples include regression and classification. Unsupervised learning, on the other hand, deals with **unlabeled data and focuses on finding patterns or structures within the data**, such as clustering or dimensionality reduction."

1. \*\*Model performance evaluation:\*\*

"**I evaluate model performance using appropriate metrics depending on the problem type. For classification tasks, I might use metrics like accuracy, precision, recall, F1-score, and ROC-AUC. For regression tasks, I typically use metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), or R-squared. I also utilize techniques like cross-validation to assess model generalization performance."**

1. \*\*Experience with natural language processing (NLP) techniques:\*\*

"I've worked on projects involving text data where I applied various NLP techniques such as tokenization, stemming, lemmatization, and named entity recognition. I've also used word embeddings like Word2Vec or GloVe for semantic analysis and built models for sentiment analysis, text classification, and topic modeling using techniques like TF-IDF or word frequency."

1. \*\*Cross-validation and its importance:\*\*

"Cross-validation is a technique used to assess how well a predictive model will generalize to an independent dataset. **It involves splitting the dataset into multiple subsets, training the model on a portion of the data, and validating it on the remaining portion. Cross-validation helps to detect overfitting and provides a more reliable estimate of model performance.**"

1. \*\*Handling imbalanced datasets:\*\*

"I've encountered imbalanced datasets in classification tasks, where one class is significantly more prevalent than others. To address this, I've used techniques such as oversampling the minority class (e.g., SMOTE), undersampling the majority class, or using algorithmic approaches like cost-sensitive learning or **ensemble methods like XGBoost or RandomForest, which can handle class imbalances**."

1. \*\*Overfitting and prevention:\*\*

"Overfitting occurs when a model learns the training data too well to the extent that it performs poorly on unseen data. To prevent overfitting, I employ techniques like **cross-validation**, regularization (e.g., L1, L2 regularization), early stopping during model training, and using simpler models. I also keep a close eye on model complexity and monitor performance metrics on both training and validation datasets."

1. \*\*Interpreting clustering algorithm results:\*\*

"When interpreting clustering algorithm results, I first examine the cluster centroids or prototypes to understand the characteristics of each cluster. I then visualize the clusters using techniques like t-SNE or PCA to gain insights into the data's underlying structure. Additionally, I may analyze cluster purity or silhouette scores to assess the quality of the clustering."

1. \*\*Experience with time series data:\*\*

"I've worked extensively with time series data in various domains such as finance, healthcare, and IoT. My approach involves time series decomposition to identify trends, seasonality, and noise. I also use techniques like ARIMA, SARIMA, or Prophet for forecasting and anomaly detection. **Feature engineering plays a crucial role, where I create lagged variables or rolling statistics to capture temporal dependencies**."

1. \*\*Deploying machine learning models into production:\*\*

"I have experience deploying machine learning models into production environments using **containerization technologies like Docker** and orchestration tools like Kubernetes. I create RESTful APIs using frameworks like Flask or FastAPI for model inference. I also integrate model monitoring and logging to track model performance and drift over time, ensuring the model's reliability and scalability."

1. \*\*Programming languages and tools proficiency:\*\*

"I'm proficient in programming languages like Python and R for data manipulation, analysis, and modeling. I use libraries such as Pandas, NumPy, scikit-learn, TensorFlow, and PyTorch for various data science tasks. I'm also comfortable working with SQL for querying databases and have experience with data visualization tools like Matplotlib, Seaborn, and Plotly."

1. \*\*Staying updated with advancements in data science:\*\*

"I stay updated with the latest advancements in data science through various channels such as research papers, online courses, conferences, and professional networks like LinkedIn and GitHub. I actively participate in online communities and forums like Stack Overflow and Reddit's data science community to exchange ideas and stay informed about emerging trends and best practices."

1. \*\*Handling large volumes of data:\*\*

"In projects involving large volumes of data, I leverage distributed computing frameworks like Apache Spark for scalable data processing and analysis. I design data pipelines using technologies like Apache Airflow or Luigi for workflow orchestration and monitoring. I also optimize data storage and retrieval using technologies like Hadoop HDFS or cloud-based solutions like Amazon S3 or Google Cloud Storage."

1. \*\*Handling missing data:\*\*

"When handling missing data, I first assess the extent and pattern of missingness in the dataset. Depending on the scenario**, I may employ techniques such as imputation using mean, median, or mode for numerical data, or using forward-fill or backward-fill for time series data**. In cases where missingness is significant, I may consider advanced imputation methods like K-nearest neighbors (KNN) or predictive modeling-based imputation."