

◇ 1. What are the differences between TRUNCATE and DELETE in SQL?

Answer:

- TRUNCATE is a DDL command, whereas DELETE is a DML command.
 - TRUNCATE removes all records from a table without logging individual row deletions and **cannot be rolled back** (unless used in a transaction with savepoints in some RDBMS).
 - DELETE removes selected rows, logs each deletion, and **can be rolled back**.
-

◇ 2. How does the SAVEPOINT command improve transaction control in PL/SQL?

Answer: SAVEPOINT allows partial rollbacks in a transaction. If you have multiple operations, you can mark checkpoints. If an error occurs later, ROLLBACK TO savepoint_name; reverts to that checkpoint rather than rolling back the entire transaction.

◇ 3. Explain how PL/SQL exception handling works with an example.

Answer: PL/SQL exceptions are runtime errors. Syntax:

```
sql
BEGIN
  -- Some code
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('No data found!');
END;
```

When a known error like NO_DATA_FOUND occurs, the corresponding block is executed instead of crashing the program.

◇ 4. Can you use IF-ELSE inside a loop in PL/SQL? Demonstrate with an example.

Answer: Yes, conditional statements can be nested within loops:

```
sql
FOR i IN 1..5 LOOP
  IF MOD(i, 2) = 0 THEN
    DBMS_OUTPUT.PUT_LINE(i || ' is even');
  ELSE
    DBMS_OUTPUT.PUT_LINE(i || ' is odd');
  END IF;
END LOOP;
```

◇ 5. How does a PL/SQL trigger differ from a stored procedure?

Answer:

- A **trigger** is an automatic action fired in response to table events (e.g., AFTER INSERT), while a **procedure** must be called explicitly.
 - Triggers are useful for **auditing**, **enforcing constraints**, and **automatic data changes**.
-

◇ 6. What are IN, OUT, and IN OUT parameters in PL/SQL procedures?

Answer:

- IN: Read-only input value.
- OUT: Used to return values from the procedure.
- IN OUT: Can receive and return values.

```
sql
PROCEDURE update_salary(emp_id IN NUMBER, new_salary IN OUT NUMBER)
```

◇ 7. Explain the difference between CHAR and VARCHAR2 in PL/SQL.

Answer:

- CHAR(n) stores fixed-length data. If less, it pads with spaces.
 - VARCHAR2(n) stores variable-length strings, saving space.
-

◇ 8. What is the output of this PL/SQL code: MOD(10, 3)?

Answer: The MOD function returns the **remainder**, so MOD(10, 3) = 1.

◇ 9. What is the purpose of the DBMS_OUTPUT.PUT_LINE procedure?

Answer: It prints output to the console in SQL Developer or similar IDEs for debugging and showing results.

◇ 10. Write a PL/SQL procedure to calculate factorial of a number using a loop.

Answer:

sql

```
DECLARE
    num NUMBER := 5;
    fact NUMBER := 1;
BEGIN
    FOR i IN 1..num LOOP
        fact := fact * i;
    END LOOP;
    DBMS_OUTPUT.PUT_LINE('Factorial is ' || fact);
END;
```

◇ 11. What are string functions in PL/SQL? Name a few.

Answer:

- LENGTH, SUBSTR, INSTR, REPLACE, UPPER, LOWER, CONCAT. Used for manipulating character data.
-

◇ 12. What is the difference between CASE and IF-ELSE in PL/SQL?

Answer:

- IF-ELSE is more flexible with complex conditions.
 - CASE is cleaner for multi-value comparisons and improves readability.
-

◇ 13. How do you create and call a stored procedure in PL/SQL?

Answer:

sql

```
CREATE OR REPLACE PROCEDURE greet IS
BEGIN
    DBMS_OUTPUT.PUT_LINE('Hello!');
END;

-- Call
BEGIN
    greet;
END;
```

◆ 14. Explain the use of FOR, WHILE, and simulated DO WHILE loops.

Answer:

- FOR: Executes fixed iterations.
 - WHILE: Executes while condition is true.
 - DO WHILE: Simulated using LOOP ... EXIT WHEN.
-

◆ 15. What is the purpose of the REVOKE statement?

Answer: It removes previously granted permissions from users/roles. For example: REVOKE INSERT ON employees FROM hr_user;

◆ 16. What happens when you try to drop a table that has dependencies?

Answer: The database will throw an error unless CASCADE CONSTRAINTS is used, or you manually remove dependent objects first.

◆ 17. Write a PL/SQL block that checks if a number is prime.

Answer:

sql

```
DECLARE
    num NUMBER := 7;
    is_prime BOOLEAN := TRUE;
BEGIN
    FOR i IN 2..num-1 LOOP
        IF MOD(num, i) = 0 THEN
            is_prime := FALSE;
            EXIT;
        END IF;
    END LOOP;
    IF is_prime THEN
        DBMS_OUTPUT.PUT_LINE(num || ' is Prime');
    ELSE
        DBMS_OUTPUT.PUT_LINE(num || ' is not Prime');
    END IF;
END;
```

◆ 18. What is a trigger and give a real-world use case.

Answer: A trigger is an automatic procedure fired by DML operations. Example: Audit table changes by recording updates in a log table.

◆ 19. Explain the difference between PRIMARY KEY and UNIQUE.

Answer: Both enforce uniqueness, but:

- PRIMARY KEY does not allow nulls and is one per table.
 - UNIQUE allows one or more nulls and multiple per table.
-

◆ 20. How can you update a specific row's salary conditionally in SQL?

Answer:

sql

```
UPDATE employees
SET salary = salary * 1.10
WHERE employee_id = 2;
```

◇ 21. Describe JOIN types in SQL and their differences.

Answer:

- INNER JOIN: Matches rows from both tables.
 - LEFT JOIN: Includes all rows from left, matching from right.
 - RIGHT JOIN: Opposite of LEFT.
 - FULL JOIN: Includes all rows from both.
-

◇ 22. What is the use of GROUP BY and HAVING in SQL?

Answer:

- GROUP BY aggregates rows by column.
- HAVING filters aggregated results.

sql

```
SELECT department_id, AVG(salary)
FROM employees
GROUP BY department_id
HAVING AVG(salary) > 60000;
```

◇ 23. How can you insert multiple records in one SQL statement?

Answer:

sql

```
INSERT INTO employees VALUES
(1, 'John', 'Doe', ...),
(2, 'Jane', 'Smith', ...);
```

◇ 24. What are constants in PL/SQL and why use them?

Answer: Constants are variables whose values cannot be changed after initialization. They improve code clarity and prevent accidental changes.

◇ 25. What is Tableau and how does it integrate with DW?

Answer: Tableau is a BI tool used to visualize warehouse data using charts and dashboards. It connects to data warehouses to:

- Build reports
- Perform live or extract-based queries
- Enable business users to interpret multidimensional data.

◇ 1. What is the ARFF format and why is it used in WEKA?

Answer: ARFF (Attribute-Relation File Format) is a plain-text format developed for WEKA. It consists of:

- A **header section**: defines relation name and attributes.
 - A **data section**: comma-separated values corresponding to attributes. Used because WEKA reads this format directly and supports nominal, numeric, string, and date attributes.
-

◇ 2. What is the function of the 'Explorer' in WEKA?

Answer: Explorer is WEKA's main GUI that provides tabs like:

- Preprocess: Clean/prepare data.
 - Classify: Apply ML algorithms.
 - Cluster: Perform clustering.
 - Associate: Run association rule mining.
 - Visualize: Explore data graphically. It helps step-by-step execution of data mining tasks.
-

◇ 3. Differentiate between 'Preprocess' and 'Classify' panels in WEKA.

Answer:

- **Preprocess**: Data loading, filtering, transforming, and cleaning.
 - **Classify**: Apply ML algorithms to build models, evaluate them using metrics (accuracy, confusion matrix, etc.).
-

◇ 4. What is discretization, and why is it useful in data mining?

Answer: Discretization converts continuous data into categorical data (bins/intervals). Useful because:

- Some algorithms (like Naïve Bayes) perform better with discrete input.
 - Reduces overfitting and noise.
 - Improves interpretability.
-

◇ 5. What is the purpose of data preprocessing in WEKA?

Answer: Data preprocessing ensures data quality before modeling. Tasks include:

- Removing missing values
 - Normalization
 - Attribute selection
 - Type conversion This improves accuracy and performance of models.
-

◇ 6. What are the key features of the J48 algorithm?

Answer: J48 is WEKA's implementation of the C4.5 algorithm.

- It builds decision trees using info gain ratio.
 - Handles continuous and categorical features.
 - Supports pruning to avoid overfitting.
 - Generates interpretable if-then rules.
-

◇ 7. How does Naïve Bayes algorithm work in WEKA?

Answer: Naïve Bayes uses Bayes' theorem: $P(\text{Class}|\text{Features}) = P(\text{Features}|\text{Class}) * P(\text{Class}) / P(\text{Features})$ It assumes feature independence. In WEKA:

- Load dataset → Classify → Choose Naïve Bayes → Run WEKA outputs prior and conditional probabilities.
-

◇ 8. Describe the steps to install WEKA.

Answer:

1. Go to <https://www.cs.waikato.ac.nz/ml/weka/>
 2. Choose version (with or without JVM)
 3. Install via .exe installer
 4. Launch → Use Explorer or Command Line interfaces.
-

◇ 9. What is the significance of the ‘Knowledge Flow’ interface in WEKA?

Answer: It provides a **drag-and-drop workflow interface** to visually build and execute data mining pipelines — similar to tools like KNIME or Orange.

◇ 10. Explain association rule mining with Apriori in WEKA.

Answer: Apriori discovers frequent itemsets and generates rules of the form: IF A AND B THEN C (Support, Confidence) Steps in WEKA:

- Load dataset → Associate panel → Choose Apriori → Run You’ll get rules with support/confidence thresholds.
-

◇ 11. What is normalization and how is it done in WEKA?

Answer: Normalization scales numerical data to a 0–1 range. In WEKA:

- Preprocess → Choose Normalize filter → Apply Prevents features with large scales from dominating models.
-

◇ 12. What does the “Visualize All” button do in WEKA?

Answer: Displays scatter plots for all attribute pairs, helping detect:

- Relationships
 - Outliers
 - Class distributions
-

◇ 13. What is the .csv file format and why is it useful in WEKA?

Answer: CSV (Comma-Separated Values) stores tabular data in plain text. Useful in WEKA:

- Easily converted to ARFF
 - Compatible with most software (Excel, Python, etc.)
 - Fast import/export
-

◇ 14. What is meant by “Training”, “Testing”, and “Validation” datasets?

Answer:

- **Training:** Used to build the model.
 - **Testing:** Used to evaluate final model accuracy.
 - **Validation:** Fine-tune model parameters during training. In WEKA: Classify tab → Test options → Choose split/percentage/cross-validation.
-

◇ 15. What is the ‘Confidence Factor’ in J48 and how does it affect pruning?

Answer: Confidence factor (default: 0.25) controls pruning:

- Low values → more pruning (less overfitting)
 - High values → less pruning (may overfit)
-

◇ 16. What is the use of the 'Experimenter' in WEKA?

Answer: It allows systematic **comparison of ML algorithms** using statistical tests like t-test, ANOVA. You can run multiple classifiers over multiple datasets and record performance.

◇ 17. How can you load a dataset in WEKA from a .csv file?

Answer:

- Go to Explorer → Preprocess → Open file
 - Choose .csv file
 - WEKA automatically converts it or prompts to convert to ARFF.
-

◇ 18. How does the K-Means algorithm work in WEKA's Cluster panel?

Answer:

- Partitions data into K clusters
 - Minimizes intra-cluster distance
 - In WEKA: Choose SimpleKMeans → Set number of clusters → Run Result: Instances grouped, centroids shown.
-

◇ 19. What is information gain and how is it used in decision trees?

Answer: It measures reduction in entropy (uncertainty) after a feature split. Used by J48 to choose the best feature for splitting at each node.

◇ 20. What does 'Evaluate on test split' mean in WEKA?

Answer: This evaluates the model's performance on unseen data, preventing overfitting. You can:

- Use percentage split (e.g., 70% train, 30% test)
 - Use separate test file
 - Use cross-validation
-

◇ 21. Why is 'Data Cleaning' essential before mining?

Answer: Dirty data leads to wrong models. Cleaning includes:

- Removing duplicates
 - Handling missing values
 - Correcting outliers Improves model accuracy and trustworthiness.
-

◇ 22. What are evaluation metrics provided by WEKA for classification?

Answer:

- Accuracy
 - Confusion Matrix
 - Precision, Recall
 - F1-Score
 - ROC AUC These help compare classifiers fairly.
-

◇ 23. Describe the process to apply filters in WEKA.

Answer:

- Go to Preprocess panel
 - Click "Choose" under Filter
 - Pick filter (e.g., `unsupervised.attribute.Normalize`)
 - Click Apply
-

◇ 24. What are some limitations of WEKA?

Answer:

- Handles small/medium datasets well; struggles with big data
 - GUI-based, less flexible than code
 - Fewer options for deep learning
 - Needs Java environment
-

◇ 25. What are some real-world applications of DMPA techniques in WEKA?

Answer:

- **Healthcare:** Predict disease risk
- **Retail:** Customer segmentation
- **Finance:** Fraud detection
- **Education:** Predict student performance
- **Marketing:** Recommendation systems

◇ 1. What is the difference between R and RStudio?

Answer:

- **R** is a programming language and environment for statistical computing.
 - **RStudio** is an IDE (Integrated Development Environment) that provides tools like console, script editor, environment viewer, and plotting window — making R easier to use.
-

◇ 2. How do you install and load multiple packages in R at once?

Answer:

r

```
install.packages(c("dplyr", "ggplot2", "readxl"))  
library(dplyr); library(ggplot2); library(readxl)  
install.packages() installs, and library() loads them into memory.
```

◇ 3. Explain the use of %/% and %% in R with example.

Answer:

- %/% returns **integer quotient**
- %% returns **remainder**

r

```
10 %/% 3 # Output: 3  
10 %% 3  # Output: 1
```

◇ 4. What are the different types of vectors in R?

Answer:

- **Numeric:** `c(1, 2.5, 3.1)`
 - **Integer:** `c(1L, 2L)`
 - **Character:** `c("a", "b")`
 - **Logical:** `c(TRUE, FALSE)`
 - **Complex:** `c(2+3i)` All elements must be of the **same type**.
-

◇ 5. What does `rev()` and `sort()` do in R?

Answer:

- `rev(x)`: Reverses a vector
- `sort(x)`: Sorts vector in ascending order

r

```
v <- c(5,3,8); rev(v) → 8 3 5; sort(v) → 3 5 8
```

◇ 6. How do you read a CSV file and find the highest salary in R?

Answer:

r

```
data <- read.csv("file.csv")  
max_salary <- max(data$salary)
```

◇ 7. How can you create a labeled matrix in R?

Answer:

r

```
m <- matrix(1:6, nrow=2)  
rownames(m) <- c("Row1", "Row2")  
colnames(m) <- c("Col1", "Col2", "Col3")
```

◇ 8. What is the use of subset() function?

Answer: It extracts rows from data frames meeting specific conditions.

r

```
subset(data, salary > 50000 & department == "IT")
```

◇ 9. Write a function to calculate the sum of digits of any number.

Answer:

r

```
sum_digits <- function(n) {  
  sum(as.numeric(strsplit(as.character(n), "")[[1]]))  
}  
sum_digits(1234) # Output: 10
```

◇ 10. How do you visualize data using ggplot2?

Answer:

r

```
library(ggplot2)  
ggplot(iris, aes(Sepal.Length, Petal.Length, color=Species)) +  
  geom_point()
```

It creates a **scatter plot** with color representing species.

◇ 11. What is the difference between geom_bar() and geom_histogram()?

Answer:

- geom_bar() is for **categorical data**
 - geom_histogram() is for **continuous data**, automatically binned
-

◇ 12. How do you normalize a vector in R?

Answer:

r

```
normalize <- function(x) {  
  (x - min(x)) / (max(x) - min(x))  
}  
normalize(c(2,4,6))
```

◇ 13. How do you split data into training and testing sets using caTools?

Answer:

r

```
library(caTools)  
split <- sample.split(data$target, SplitRatio=0.7)  
train <- subset(data, split==TRUE)  
test <- subset(data, split==FALSE)
```

◇ 14. Write a code to implement linear regression and plot the result.

Answer:

r

```
model <- lm(Salary ~ Years_Exp, data = training)  
ggplot(training, aes(Years_Exp, Salary)) +  
  geom_point() +  
  geom_smooth(method = "lm")
```

◇ 15. How do you extract column names from a data frame?

Answer:

```
r  
names(data) # Returns all column names
```

◇ 16. What is the difference between == and identical() in R?

Answer:

- == compares **element-wise**
- identical() compares **complete objects**

```
r  
c(1,2) == c(1,2)      # TRUE TRUE  
identical(c(1,2), c(1,2)) # TRUE
```

◇ 17. Explain the use of geom_smooth() in ggplot.

Answer: It adds a trend line (like regression) to scatter plots. method="lm" applies linear model fitting.

◇ 18. What is a diverging bar chart and how is it used in R?

Answer: It visualizes performance above/below average.

```
r  
mtcars$mpg_z <- scale(mtcars$mpg)  
ggplot(mtcars, aes(x=rownames(mtcars), y=mpg_z)) +  
  geom_bar(stat="identity")
```

◇ 19. How do you create a lollipop chart in ggplot2?

Answer:

```
r  
ggplot(mtcars, aes(x=rownames(mtcars), y=mpg)) +  
  geom_point() +  
  geom_segment(aes(xend=rownames(mtcars), yend=0))
```

◇ 20. How do you calculate correlation between two numeric vectors?

Answer:

```
r  
cor(x, y) # Returns Pearson correlation coefficient
```

◇ 21. What is the purpose of ggsave()?

Answer: Saves the last plot as an image:

```
r  
ggsave("plot.png", width=4, height=4, dpi=300)
```

◇ 22. How do you create a vector with alternating elements in R?

Answer:

```
r  
rep(c(4,6,3), times=10) # Repeats pattern 10 times
```

◇ 23. How do you label axes and title a ggplot?

Answer:

r

```
+ labs(x = "Sepal Length", y = "Petal Length") +  
  ggtitle("Sepal vs Petal")
```

◇ 24. What is logistic regression in R and how is it implemented?

Answer:

r

```
model <- glm(am ~ cyl + hp + wt, data=mtcars, family=binomial)  
summary(model)
```

Used for binary classification.

◇ 25. How do you create a matrix with custom row/column names?

Answer:

r

```
m <- matrix(1:6, nrow=2)  
rownames(m) <- c("A", "B")  
colnames(m) <- c("X", "Y", "Z")
```

◇ 1. What is the purpose of learning R in Applied Statistics?

Answer: R is a powerful language for **statistical computing and data visualization**. It provides:

- Built-in functions for descriptive and inferential statistics
 - Packages like `tidyverse`, `ggplot2`, `stats`
 - Reproducibility and script-based workflow It's widely used in academia and industry for data analysis.
-

◇ 2. How do you check if R and RStudio are updated to the latest version?

Answer: In RStudio:

- Go to Help → Check for Updates In R:
 - Run `sessionInfo()` to check R version Compare with the latest on <https://cran.r-project.org>
-

◇ 3. What is a sparse matrix, and how can you check it in R?

Answer: A matrix is **sparse** if most of its elements are zero. To check:

```
r
matrix <- matrix(c(0, 0, 1, 0), 2, 2)
sparsity <- sum(matrix == 0) / length(matrix)
if(sparsity > 0.5) print("Sparse Matrix")
```

◇ 4. How do you create a stem-and-leaf plot in R?

Answer:

```
r
data <- c(12, 15, 16, 17, 21, 25, 29)
stem(data)
```

It's a textual representation showing distribution.

◇ 5. What is a boxplot, and what statistical values does it show?

Answer: A boxplot visualizes:

- **Median (Q2)**
- **Q1 and Q3**
- **IQR**
- **Whiskers (min, max)**
- **Outliers** Use in R:

```
r
boxplot(data)
```

◇ 6. How do you calculate sum, mean, median, and variance of a vector?

Answer:

```
r
v <- c(1, 2, 3, 4, 5)
sum(v); mean(v); median(v); var(v)
```

◇ 7. How do you implement frequency distribution in R?

Answer:

```
r
data <- c(1,2,2,3,3,3)
table(data) # Gives frequency of each unique value
```

◇ 8. What is sampling distribution and how can it be visualized in R?

Answer: Sampling distribution shows distribution of a statistic (e.g., sample mean). Simulate:

r

```
means <- replicate(1000, mean(sample(1:100, 10)))  
hist(means)
```

◇ 9. How do you check if a matrix is symmetric?

Answer:

r

```
mat <- matrix(c(1,2,2,1), 2)  
isSymmetric(mat) # Returns TRUE
```

◇ 10. What does switch() do in R?

Answer: It replaces complex if-else blocks:

r

```
choice <- 2  
switch(choice, "One", "Two", "Three") # Output: "Two"
```

◇ 11. How do you write a function in R to calculate factorial?

Answer:

r

```
fact <- function(n) {  
  if(n==0) return(1)  
  return(n * fact(n-1))  
}
```

◇ 12. How can you generate a vector with alternating TRUE/FALSE values?

Answer:

r

```
rep(c(TRUE, FALSE), length.out = 10)
```

◇ 13. Write code to check if a number is even or odd.

Answer:

r

```
x <- 4  
if (x %% 2 == 0) print("Even") else print("Odd")
```

◇ 14. What does var() function return in R?

Answer: var() returns **sample variance** (not population variance).

◇ 15. How do you create and access a data frame in R?

Answer:

r

```
df <- data.frame(name=c("A","B"), marks=c(90,80))  
df$name # Access column  
df[1,] # Access row
```

◇ 16. What is the difference between `c()` and `list()` in R?

Answer:

- `c()` creates **atomic vectors**
- `list()` creates **heterogeneous lists**

r

```
c(1, "a", TRUE) → coerces to character  
list(1, "a", TRUE) → retains types
```

◇ 17. How do you write an R script to print odd numbers between 1 and 20?

Answer:

r

```
for(i in 1:20) {  
  if(i %% 2 != 0) print(i)  
}
```

◇ 18. How do you calculate log base 2 and square root together?

Answer:

r

```
x <- sqrt(2345)  
log2(x)
```

◇ 19. How can you use `plot()` to visualize a dataset?

Answer:

r

```
x <- 1:10; y <- x^2  
plot(x, y, type='b', col='blue')  
Displays scatter or line plot.
```

◇ 20. How do you filter rows from a data frame using logical conditions?

Answer:

r

```
subset(df, marks > 80)
```

◇ 21. How do you find the number of rows and columns in a data frame?

Answer:

r

```
nrow(df); ncol(df)
```

◇ 22. How do you generate random normal data and plot its histogram?

Answer:

r

```
data <- rnorm(1000, mean=50, sd=10)  
hist(data)
```

◇ 23. How do you calculate standard deviation in R?

Answer:

r

```
sd(c(1,2,3,4,5)) # Returns standard deviation
```

◇ 24. What does `t.test()` function do in R?

Answer: Performs a **t-test** to compare means of groups:

r

```
t.test(group1, group2)
```

Useful in hypothesis testing.

◇ 25. What's the use of `summary()` function in R?

Answer:

r

```
summary(df)
```

Gives min, max, mean, median, and quartiles for each numeric column.