SQL\_F1\_DESCRIPTIVE

# Chapter: Introduction

1. Define the relational and object relational database management systems.

Relational and object relational database management systems:

1. Relational model and object relational model
2. User-define data types and objects
3. Fully compatible with relational database
4. Supports multimedia and large objects
5. High-quality database server features
6. Write down the principles of the relational database model.
   1. Dr. E. F. Codd proposed the relational model for database systems in 1970.
   2. It is the basis for the relational database management system (RDBMS).
   3. The relational model consists of the following:
      1. Collection of objects or relations
      2. Set of operators to act on the relations
      3. Data integrity for accuracy and consistency
7. What is Relational Database?

A relational database is a collection of relations or two-dimensional tables.

1. What is data models? Write the purpose of data models.

Data Models:

Models are a cornerstone of design. Engineers build a model of a car to work out any details before putting it into production.

Purpose of Models:

Models help communicate the concepts that are in people’s minds. They can be used to do the following:

* + - Communicate
    - Categorize
    - Describe
    - Specify
    - Investigate
    - Evolve
    - Analyze
    - Imitate

1. Write down the benefits of ERD modeling.

Benefits of ER Modeling

* + - Documents information for the organization in a clear, precise format
    - Provides a clear picture of the scope of the information requirement
    - Provides an easily understood pictorial map for database design
    - Offers an effective framework for integrating multiple applications

1. Write down the ER Modeling Conventions.

ER Modeling Conventions:

* 1. Entities

To represent an entity in a model, use the following conventions:

* + - Singular, unique entity name
    - Entity name in uppercase
    - Soft box
    - Optional synonym names in uppercase within parentheses: ( )
  1. Attributes

To represent an attribute in a model, use the following conventions:

* + - Singular name in lowercase
    - Asterisk (\*) tag for mandatory attributes (i.e., values that *must* be known)
    - Letter “o” tag for optional attributes (i.e., values that *may* be known)
  1. Relationships

Each source entity {may be | must be} relationship name {one and only one | one or more} destination entity.

1. Difference between primary key and foreign key.
   1. Each row of data in a table is uniquely identified by a primary key (PK).
   2. You can logically relate data from multiple tables using foreign keys (FK).
2. Write down the Guidelines for Primary Keys and Foreign Keys

Guidelines for Primary Keys and Foreign Keys:

* + - You cannot use duplicate values in a primary key.
    - Primary keys generally cannot be changed.
    - Foreign keys are based on data values and are purely logical (not physical) pointers.
    - A foreign key value must match an existing primary key value or unique key value, or else it must be null.
    - A foreign key must reference either a primary key or a unique key column.

1. What are unique identifiers?

A unique identifier (UID) is any combination of attributes or relationships, or both, that serves to distinguish occurrences of an entity. Each entity occurrence must be uniquely identifiable.

* + - Tag each attribute that is part of the UID with a number sign: #
    - Tag secondary UIDs with a number sign in parentheses: (#)

1. Write down the Relational Database Terminology.

Terminology Used in a Relational Database

A relational database can contain one or many tables. A *table* is the basic storage structure of an RDBMS. The contents of the EMPLOYEES *table* or *relation* are the following:

1. A single *row* (or *tuple*)representing all data required for a particular employee. Each row in a table should be identified by a primary key, which permits no duplicate rows.
2. A *column* orattributecontaining the employee number. The employee number identifies a *unique* employee in the EMPLOYEES table.
3. A column that is not a key value. A column represents one kind of data in a table.
4. A column containing the department number, which is also a *foreign key*. A foreign key is a column that defines how tables relate to each other. A foreign key refers to a primary key or a unique key in the same table or in another table.
5. A *field* can be found at the intersection of a row and a column. There can be only one value in it.
6. A field may have no value in it. This is called a null value.
7. What is SQL?

Structured Query Language (SQL) is:

* 1. The ANSI standard language for operating relational database
  2. Efficient, easy to learn, and use
  3. Functionally complete

1. Write down the both ANSI and ISO accepted SQL statements.

Both ANSI and ISO have accepted SQL as the standard language for relational databases:

* 1. Data manipulation language (DML)

SELECT

INSERT

UPDATE

DELETE

MERGE

* 1. Data definition language (DDL)

CREATE

ALTER

DROP

RENAME

TRUNCATE

COMMENT

* 1. Transaction control

COMMIT

ROLLBACK

SAVEPOINT

* 1. Data control language (DCL)

GRANT

REVOKE

# Chapter: 1

1. Write down the capabilities of SQL statement?

Capabilities of SQL statements:

A select statement retrieves information from the database. With a select statement, you can use the following capabilities:

a) Projection

b) Selection

c) Joining

1. Write down the Basic SELECT Statement Conventions.

In its simplest form, a SELECT statement must include the following:

* 1. SELECT identifies the columns to be displayed
  2. FROM identifies the table containing those columns

1. Write down the basic SQL Statement syntax?

In the Syntax:

SELECT \* / {[DISYINCT] column /expression [alias]…..} FROM table;

1. What is a guideline/simple rule of writing SQL Statement?

Using the following simple rules and guidelines, you can construct valid statements that are both easy to read and easy to edit:

* + - SQL statements are not case-sensitive (unless indicated).
    - SQL statements can be entered on one or many lines.

• Keywords cannot be split across lines or abbreviated.

* + - Clauses are usually placed on separate lines for readability and ease of editing.

• Indents should be used to make code more readable.

* Keywords typically are entered in uppercase

1. Write down the default column heading rule in SQL Developer and SQL \* Plus.

SQL Developer:

* 1. Default heading alignment Left-aligned
  2. Default heading display Uppercase

SQL\*Plus:

1. Character and Date column headings are Left-aligned
2. Number column headings are Right-aligned
3. Default heading display Uppercase
4. What is Null value?

A null is a value that is unavailable, unassigned, unknown, or inapplicable. A null is not the same as a zero or a space. Zero is a number, and a space is a character.

1. What is a column alias?

A column alias:

* + Renames a column heading
  + Is useful with calculations
  + Immediately follows the column name (There can also be the optional AS keyword between the column name and alias.)
  + Requires double quotation marks if it contains spaces or special characters or if it is case-sensitive

1. Define the Concatenation Operator.

A concatenation operator:

* + Links columns or character strings to other columns
  + Is represented by two vertical bars (||)
  + Creates a resultant column that is a character expression

1. Define the Literal Character Strings.
   1. A literal is a character, a number, or a date that is included in the SELECT statement.
   2. Date and character literal values must be enclosed by single quotation marks.
   3. Each character string is output once for each  
      row returned.

# Chapter: 2

1. Write down the use of WHERE clause.
   1. Restrict the rows that are returned by using the WHERE clause.
   2. The WHERE clause follows the FROM clause.
2. Write down the comparison operators.

|  |  |
| --- | --- |
| Operator | Meaning |
| = | Equal to |
| > | Greater than |
| >= | Greater than or equal to |
| < | Less than |
| <= | Less than or equal to |
| <> | Not equal to |
| BETWEEN ...AND... | Between two values (inclusive) |
| IN(set) | Match any of a list of values |
| LIKE | Match a character pattern |
| IS NULL | Is a null value |

1. Write down the use of LIKE operator.
   1. Use the LIKE condition to perform wildcard searches of valid search string values.
   2. Search conditions can contain either literal characters or numbers:
      1. % denotes zero or many characters.
      2. \_ denotes one character.
2. Write down the logical operations used in SQL.

Three logical operators are available in SQL:

• AND

• OR

• NOT

1. Write down the use of ORDER BY clause in SQL.
   1. Sort retrieved rows with the ORDER BY clause:

ASC: ascending order, default

DESC: descending order

* 1. The ORDER BY clause comes last in the SELECT statement

1. What are the rules of operator president?

|  |  |
| --- | --- |
| Operator | Meaning |
| 1 | Arithmetic operators |
| 2 | Concatenation operator |
| 3 | Comparison conditions |
| 4 | IS [NOT] NULL, LIKE, [NOT] IN |
| 5 | [NOT] BETWEEN |
| 6 | Not equal to |
| 7 | NOT logical condition |
| 8 | AND logical condition |
| 9 | OR logical condition |

1. Write down the use of Substitution Variables.
   1. Use SQL\*Plus substitution variables to:
      1. Temporarily store values with single-ampersand (&) and double-ampersand(&&) substitution
   2. Use substitution variables to supplement the following:
      1. WHERE conditions
      2. ORDER BY clauses
      3. Column expressions
      4. Table names
      5. Entire SELECT statements

# Chapter: 3

1. What are the types of SQL Functions?

There are two types of functions:

* + - Single-row functions: Return one result per row
    - Multiple-row functions: Return one result per set of rows

1. What are the types of Single-Row Functions?

There are different types of single-row functions. This lesson covers the following ones:

* + - Character
    - Number
    - Date
    - Conversion
    - General

1. Write down the use of Single-Row Functions.

Single-row functions are used to:

* + Manipulate data items
  + Accept arguments and return one value

1. What is SYSDATE?

SYSDATE is a function that returns the current database server date and time.

# Chapter: 4

1. What are the types of Data type conversion?

* Implicit data type conversion: This data type conversion can be done *implicitly* by the Oracle server.
* Explicit data type conversion: This data type conversion can be done *explicitly* by the user.

1. Write down the guidelines for format model.

The format model:

* + Must be enclosed by single quotation marks
  + Is case-sensitive
  + Can include any valid date format element
  + Has an fm element to remove padded blanks or suppress leading zeros
  + Is separated from the date value by a comma

1. What is Nesting functions?
   1. Single-row functions can be nested to any level.
   2. Nested functions are evaluated from deepest level to the least deep level.
2. What is General functions?

* The General functions work with any data type and pertain to using nulls.

1. What is Conditional Expressions?
   1. Provide the use of IF-THEN-ELSE logic within a SQL statement
   2. Use two methods:
      1. CASE expression
      2. DECODE function

# Chapter: 5

1. What is Group Function? Write down the types of Group Function?

Group functions: Group functions operate on sets of rows to give one result per group.

Types of Group Functions:

* + AVG
  + COUNT
  + MAX
  + MIN
  + STDDEV
  + SUM
  + VARIANCE

1. Write down the guidelines for using group functions:
   * + DISTINCT makes the function consider only nonduplicate values.
     + The data types for the functions with an expr argument may be CHAR, VARCHAR2, NUMBER, or DATE.
     + All group functions ignore null values. To substitute a value for null values, use the NVL, NVL2, or COALESCE functions.
2. Describe the formats of COUNT function.

The COUNT function has three formats:

• COUNT(\*)returns the number of rows in a table.

• COUNT(*expr*) returns the number of rows with non-null values for *expr.*

• COUNT(*DISTINCT expr*) returns the number of unique rows with non-null values for *expr.*

1. Write down the relation between Group Functions and Null Values.

* All group functions ignore null values in the column.
* The NVL function forces group functions to include null values.

1. Write down the guidelines for creating groups of data.

Guidelines for creating groups of data:

* + 1. Using a WHERE clause, you can exclude rows before dividing them into groups.
    2. You must include the *columns* in the GROUP BY clause.
    3. You cannot use a column alias in the GROUP BY clause.

1. Write down the illegal queries using group functions.
   1. You cannot use the WHERE clause to restrict groups.
   2. You use the HAVING clause to restrict groups.
   3. You cannot use group functions in the WHERE clause.
2. Write down the Restricting Group Results with the HAVING Clause.

When you use the HAVING clause, the Oracle server restricts groups as follows:

1. Rows are grouped.
2. The group function is applied.
3. Groups matching the HAVING clause are displayed.

# Chapter: 6

1. Write down the types of joins with the SQL: 1999 standard.

Joins that are compliant with the SQL: 1999 standard include the following:

* + - Natural joins:

1. NATURAL JOIN clause
2. USING clause
3. ON clause
   * + OUTER joins:
4. LEFT OUTER JOIN
5. RIGHT OUTER JOIN
6. FULL OUTER JOIN
   * + Cross joins
7. Write down the guidelines for using table aliases?

* Table aliases can be up to 30 characters in length, but shorter aliases are better than longer ones.
* If a table alias is used for a particular table name in the FROM clause, then that table alias must be substituted for the table name throughout the SELECT statement.
* Table aliases should be meaningful.
* The table alias is valid for only the current SELECT statement.

1. Write down the difference between Equijoins and Non-equijoins.

An equijoin combines two or more tables based on a column that is common to the tables. Frequently, this type of join involves primary and foreign key complements. Equijoins are also called *simple joins* or *inner joins*.

A non-equijoin is a join condition containing something other than an equality operator.Other conditions can be used, but BETWEEN is the simplest.

1. What is Self-join?

Sometimes you need to join a table to itself. To find the name of each employee’s manager, you need to join the EMPLOYEES table to itself, or perform a self join.

1. Write about INNER Versus OUTER Joins.
   1. In SQL:1999, the join of two tables returning only matched rows is called an inner join.
   2. A join between two tables that returns the results of the inner join as well as the unmatched rows from the left (or right) tables is called a left (or right) outer join.
   3. A join between two tables that returns the results of an inner join as well as the results of a left and right join is a full outer join.
2. What is Cartesian product?
   1. A Cartesian product is formed when:
      1. A join condition is omitted
      2. A join condition is invalid
      3. All rows in the first table are joined to all rows in the second table
   2. To avoid a Cartesian product, always include a valid join condition.

# Chapter: 7

1. What is subquery? Where subqueries could be used?

* A subquery is a SELECT statement that is embedded in a clause of another SELECT statement
* The subquery (inner query) executes once before the main query (outer query).
* The result of the subquery is used by the main query.

You can place the subquery in a number of SQL clauses, including the following:

* SELECT *select\_list*
* FROM *table*
* WHERE *condition*
* GROUP BY *expression\_1*
* HAVING *expression\_2*

1. Write down the guidelines for using subqueries.

* Enclose subqueries in parentheses.
* Place subqueries on the right side of the comparison condition.
* The ORDER BY clause in the subquery is not needed unless you are performing Top-N analysis.
* Use single-row operators with single-row subqueries, and use multiple-row operators with multiple-row subqueries.

1. What are the types of subqueries?

* Single-row subqueries: Queries that return only one row from the inner SELECT statement. And use =, >, >=, <, <=, <> operators.
* Multiple-row subqueries: Queries that return more than one row from the inner SELECT statement. And use IN, ANY, ALL, EXISTS operators.

# Chapter: 8

1. What is Set operators? What are the types of Set operators?

The set operators combine the results of two or more component queries into one result. Queries containing set operators are called *compoundqueries*.

Types of Set Operators:

* UNION: return all rows from multiple tables and eliminate any duplicate rows.
* UNION ALL: return all rows from multiple queries.
* INTERSECT: return all rows that are common to multiple queries.
* MINUS:return rows returned by the first query that are not present in the second query (the first SELECT statement MINUS the second SELECT statement).

1. Write down the guidelines for using Set operators.
   1. The expressions in the SELECT lists must match in number.
   2. The datatype of each column in the second query must match the datatype of its corresponding column in the first query.
   3. Parentheses can be used to alter the sequence of execution.
   4. ORDER BY clause can appear only at the very end of the statement.
2. Write down the guidelines for using UNION operators.
   * + The number of columns and the data types of the columns being selected must be identical in all the SELECT statements used in the query. The names of the columns need not be identical.
     + UNION operates over all of the columns being selected.
     + NULL values are not ignored during duplicate checking.
     + The IN operator has a higher precedence than the UNION operator.
     + By default, the output is sorted in ascending order of the first column of the SELECT clause.
3. Write down the guidelines for using UNION ALL operators.

The guidelines for UNION and UNION ALL are the same, with the following two exceptions that pertain to UNION ALL:

* + - Unlike UNION, duplicate rows are not eliminated and the output is not sorted by default.
    - The DISTINCT keyword cannot be used.

1. Write down the guidelines for using INTERSECT operators.
   * + The number of columns and the data types of the columns being selected by the SELECT statements in the queries must be identical in all the SELECT statements used in the query. The names of the columns need not be identical.
     + Reversing the order of the intersected tables does not alter the result.
     + INTERSECT does not ignore NULL values.

# Chapter: 9

1. What is transaction? When does executed a DML statement?

Transaction: A *transaction* consists of a collection of DML statements that form a logical unit of work.

A DML statement is executed when:

* + Add new rows to a table
  + Modify existing rows in a table
  + Remove existing rows from a table

1. Write down the convention of Inserting New Rows.

* Insert a new row containing values for each column.
* List values in the default order of the columns in the table.
* Optionally, list the columns in the INSERT clause.
* Enclose character and date values in single quotation marks.

1. Write down the convention of Copying Rows from Another Table.
   1. Write your INSERT statement with a subquery:
   2. Do not use the VALUES clause.
   3. Match the number of columns in the INSERT clause to those in the subquery.
2. Write down the convention of Updating Rows in a Table.

* Specific row or rows are modified if you specify the WHERE clause
* All rows in the table are modified if you omit the WHERE clause

1. Write down the difference between delete and truncate.

I have summarized main difference between Truncate and Delete command in Oracle.

|  |  |  |
| --- | --- | --- |
| Hints | Truncate | Delete |
| Statement type | DDL command | DML command |
| Rollback | Cannot be rolled back | Can be rolled back |
| Commit | Associated with commit | Need to commit explicitly after |
| Grants | Truncate cannot be without using a DROP ANY TABLE grant. | Delete can be granted on a table to another user or role |
| Foreign keys | Cannot run truncate for a parent table with constraints | Delete can be applied after deleting the child records |
| Execution time | Generally executes faster than delete | Generally executes slower than truncate |
| Space reclamation | Truncate recovers space | Delete does not recover space |

1. What is a database transactions?

A database transaction consists of one of the following:

* + DML statements that constitute one consistent change to the data
  + One DDL statement
  + One data control language (DCL) statement

1. When Does a Transaction Start and End?
   1. Begin when the first DML SQL statement is executed
   2. End with one of the following events:
      1. A COMMIT or ROLLBACK statement is issued.
      2. A DDL or DCL statement executes (automatic commit).
      3. The user exits SQL\*Plus.
      4. The system crashes.
2. Write down the Advantages of COMMIT and ROLLBACK Statements.

With COMMIT and ROLLBACK statements, you can:

* + Ensure data consistency
  + Preview data changes before making changes permanent
  + Group logically related operations

1. When does implicit transaction processing?
   1. An automatic commit occurs under the following circumstances:
      1. DDL statement is issued
      2. DCL statement is issued
      3. Normal exit from SQL\*Plus, without explicitly issuing COMMIT or ROLLBACK statements
   2. An automatic rollback occurs under an abnormal termination of SQL\*Plus or a system failure.

# Chapter: 10

1. Write down the naming rules and guidelines.

Table names and column names:

* 1. Must begin with a letter
  2. Must be 1–30 characters long
  3. Must contain only A–Z, a–z, 0–9, \_, $, and #
  4. Must not duplicate the name of another object owned by the same user
  5. Must not be an Oracle server reserved word

Naming Guidelines:

Use descriptive names for tables and other database objects.

1. What are the permissions you have for creating table?

To create a table, a user must have the CREATE TABLE privilege and a storage area in which to create objects.

1. Write down the guidelines for using data types.

Guidelines:

* A LONG column is not copied when a table is created using a subquery.
* A LONG column cannot be included in a GROUP BY or an ORDER BY clause.
* Only one LONG column can be used per table.
* No constraints can be defined on a LONG column.
* You might want to use a CLOB column rather than a LONG column.

1. What are the valid types of constraints?

The following constraint types are valid:

* + 1. NOT NULL
    2. UNIQUE
    3. PRIMARY KEY
    4. FOREIGN KEY
    5. CHECK

1. Write down the guidelines for using foreign key.

Guidelines:

A foreign key value must match an existing value in the parent table or be NULL.

Foreign keys are based on data values and are purely logical, rather than physical, pointers.

1. Write down the guidelines for creating a table by using a subquery.
   1. Match the number of specified columns to the number of subquery columns.
   2. Define columns with column names and  
      default values.
2. What are the uses of ALTER TABLE statement?

Use the ALTER TABLE statement to:

* + Add a new column
  + Modify an existing column
  + Define a default value for the new column
  + Drop a column
  + Rename a column
  + Change table to read-only status

# Chapter: 11

1. What is view? Write down the advantages of views.

A view is a logical table based on a table or another view. A view contains no data of its own but is like a window through which data from tables can be viewed or changed.

Advantages of Views:

* To restrict data access
* To make complex queries easy
* To provide data independence
* To present different views of the same data

1. Write down the difference between Simple Views and Complex Views.

|  |  |  |
| --- | --- | --- |
| Feature | Simple Views | Complex Views |
| Number of tables | One | One or more |
| Contain functions | No | Yes |
| Contain groups of data | No | Yes |
| DML operations through a view | Yes | Not always |

1. What is sequence?

A sequence:

* Can automatically generate unique numbers
* Is a sharable object
* Can be used to create a primary key value
* Replaces application code
* Speeds up the efficiency of accessing sequence values when cached in memory

1. How Are Indexes Created?
   1. Automatically: A unique index is created automatically when you define a PRIMARY KEY or UNIQUE constraint in a table definition.
   2. Manually: Users can create nonunique indexes on columns to speed up access to the rows.
2. When to you create an index?

* A column contains a wide range of values
* A column contains a large number of null values
* One or more columns are frequently used together in a WHERE clause or a join condition
* The table is large and most queries are expected to retrieve less than 2% to 4% of the rows in the table

1. When you do not create an index?

* The columns are not often used as a condition in the query
* The table is small or most queries are expected to retrieve more than 2% to 4% of the rows in the table
* The table is updated frequently
* The indexed columns are referenced as part of an expression