talentech

Table Operation

(Data Definition Language)

Data Types

- Character Data types
  - Fixed length
  - Variable length
- Numeric Data Types
- Date and Time data Types
- Large Object data Types

Data Types

- **Character Data types**
  - **Fixed length:** The CHAR data type stores fixed-length character literals. When creating a CHAR column in a table, you must specify a string length between 1 and 2000 bytes for the CHAR column.
  - **Variable length:** The VARCHAR2 data type stores variable-length character literals. When creating a VARCHAR2 column in a table, you must specify a string length between 1 and 4000 bytes for the VARCHAR2 column.

Data Types

- **Numeric Data Types**
  - Use the NUMBER data type to store integers and real numbers in a fixed-point or floating-point format.
- **Date and Time data Types**
  - **DATE** – Use the DATE data type to store dates and Times in a table. For example, a column to hold the date that an employee is hired can by defined as a DATE data type.
  - **TIMESTAMP** – Use the TIMESTAMP data type to store values that are precise to fractional seconds. An application that must decide which of two events occurred first might use TIMESTAMP.
- **Large Object data Types**
  - **LOB**

Table Operation

- CREATE TABLE (table name)
- INSERT INTO (table name)
- UPDATE (table name)
- ALTER TABLE (table name)
- DELETE FROM (table name)
- DROP (table name)
- TRUNCATE

Creating a Table

- Use the CREATE TABLE statement to create a table. The simple Syntax is as follows:

```
CREATE [GLOBAL TEMPORARY] TABLE table_name (
column_name type [CONSTRAINT constraint_def DEFAULT default_exp]
[, column_name type [CONSTRAINT constraint_def DEFAULT default_exp]...])
```

- Example:

```
CREATE TABLE students (
student_id VARCHAR2(10) PRIMARY KEY,
last_name VARCHAR2(25) NOT NULL,
first_name VARCHAR2(25) NOT NULL,
Street VARCHAR2(25),
city VARCHAR2(25),
state CHAR(2),
zip_code CHAR(5),
expected_salary NUMBER(6) );
```

Adding Rows - INSERT

- Use the INSERT statement to add rows to a table. You can specify the following information in an INSERT statement:
  - The table into which the row is to be inserted
  - A list of columns for which you want to specify column values
  - A list of values to store in the specified columns
- Example:

```
INSERT INTO customers (  customer_id, first_name, last_name, dob, phone)
VALUES (  6, 'Fred', 'Brown', '01-JAN-1970', '800-555-1215');
```

Modifying Rows - UPDATE

- Use the UPDATE statement to change rows in a table. When you typically use the UPDATE statement, you specify the following information:
  - The table containing the rows that are to be changed
  - A WHERE clause that specifies the rows that are to be changed
  - A list of column names, along with their new values, specified using the SET clause
- Example

```
UPDATE customers
SET last_name = 'Orange'WHERE customer_id = 2;
```

Altering a Table

- Use ALTER TABLE statement to change the structure of a table. Use this statement to perform tasks such as:
  - Add, modify, or drop a column
  - Add or drop a constraint
  - Enable or disable a constraint
- Example: Adding a Column

```
ALTER TABLE students
ADD start_date DATE DEFAULT SYSDATE NOT NULL;
```

Removing Rows - DELETE

- Use the DELETE statement to remove rows from a table.
- Specify a WHERE clause that limits the rows
- Without a WHERE statement, *all* the rows will be deleted.
- Examples

```
DELETE FROM customers
WHERE customer_id = 2;
```

Truncating a Table

- To truncate a table use the TRUNCATE statement.
- This removes all the rows from a table
- It resets the storage area for a table.
- 300 times faster than DELETE statement
- Syntax:

```
TRUNCATE TABLE table name;
```

- Example:

```
TRUNCATE TABLE test1;
```

Dropping a Table

- To drop (remove completely) a table from the database use the DROP TABLE statement.
- It removes the structure (physical) of the table.
- Syntax

```
DROP TABLE table name;
```

- The following example drops the test2 table:

```
DROP TABLE test2;
```

EXERCISE # 1 (TABLES)

Create a table name Students with Student Id, Last name, First Name, and Phone Number

Insert 3 sets of data into the table created

Update the second record with different value

Add a new Column to the Table

Delete the first row

Drop the table

Index

- Index in SQL is created on existing tables to retrieve the rows quickly.
- When there are thousands of records in a table, retrieving information will take a long time. Therefore indexes are created on columns which are accessed frequently, so that the information can be retrieved quickly
- Indexes can be created on a single column or a group of columns.
- When a index is created, it first sorts the data and then it assigns a ROWID for each row.

Index – Cont'd

- Syntax of Index

```
CREATE INDEX index_name
ON table_name (column_name1,column_name2...);
```

- **Note:** Even though SQL indexes are created to access the rows in the table quickly, they slow down DML operations like INSERT, UPDATE, DELETE on the table, because the indexes and tables both are updated along when a DML operation is performed. So use indexes only on columns which are used to search the table frequently.

Views

- In SQL, a view is a virtual table based on the result-set of an SQL statement.
- A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database.
- You can add SQL functions, WHERE, and JOIN statements to a view and present the data as if the data were coming from one single table.
- Create View Syntax:

```
CREATE VIEW view_name AS
SELECT column_name(s)
FROM table_name
WHERE condition
```

View Vs. Table

- A view doesn't have any physical existence, but a table does
- A view is used to restrict access to table data
- A view may help to reduce the numbers of joining while writing complex queries
- Because a view is a virtual table, it does not require any spaces

Database Testing Practices

Basics database testing concepts.

Job Description – Back-end Test

- **Senior QA Analyst/Engineer (SQL backend testing)**
- Well-established company in Columbia, MD is looking for a Senior Quality Assurance Analyst. This position will be responsible for the quality of the software applications, specifically on the company's system by implementing testing procedures. This position will also be responsible for participating in all testing activities to meet overall software release goals, for designing test cases to requirements and generating traceability coverage to test plans, and developing automation test scripts and test tools.
- Requirements:
- At least 2 years experience testing **SQL back-end**
- At least 2 years experience writing test cases and test plans
- Experience with **SQL**, with the ability to **generate SQL queries** from an entity-relationship diagram
- Experience with other, related Web technologies, including HTML, CSS, JavaScript, AJAX, and XML

Job Description – Back-end Test

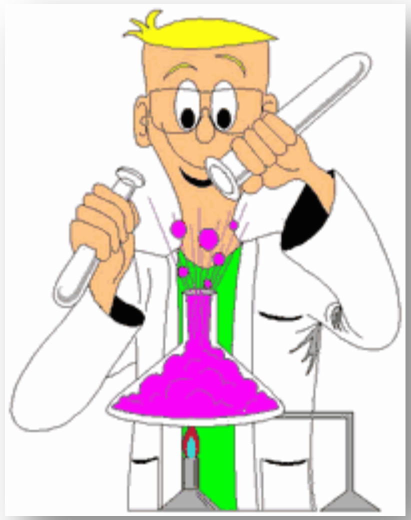
- **Job Title: Senior Backend Tester**
- **Required Skills:**
- 6 to 8 Yrs of Back **Testing** experience .
- Insurance Industry experience Preferred
- Must have well rounded **testing** skills with a firm understanding of Functional End to End **Testing**, System **Testing**, Batch Processing, User Acceptance **Testing** and Regression.
- Experience creating Test Plans and Test Cases based on functional specifications. Strong understanding of QA concepts and methodologies.
- Strong experience with software development lifecycle phases.
- Strong working knowledge of SQL Server (complex joins, data modeling, data mapping, PL/SQL   development skills are required)
- Knowledge of **ETL testing** process .
- Knowledge of relational, star schema, dimensional data models

Database Testing

- Database testing includes
  - The testing of actual data (content)
  - Database integrity
    - Ensure that data is not corrupted
    - Schemas are correct
  - Functionality testing of database application
  - SQL scripting is generally used to test databases.

Database Testing

- Database testing includes
  - The testing of actual data (content)
  - Database integrity
    - Ensure that data is not corrupted
    - Schemas are correct
  - Functionality testing of database application
  - SQL scripting is generally used to test databases.

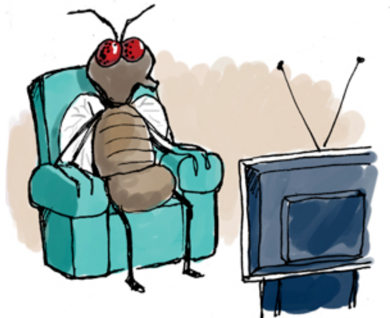


Common Errors

- Data integrity errors: Any bug that causes erroneous results to be stored.
  - We might have missing or incorrect data in records e.g, incorrect SSN in an employee record
  - We might have missing records in tables e.g an employee record missing from the employee database
  - NULL is passed to a record field that does not allow NULL

Common Errors

- Mishandling wrong data type. For example, employee salary in a record expects an integer, but receives $500 instead
- A value is too large for the size of the field
- Output errors such as retrieve incorrect data even the source is correct



Concurrency Issues

- Concurrent activities need to lock records to prevent concurrent updates and prevent data errors in the database.
  - For example, two customers should not be able to buy the same item.
  - The application design has to consider when to lock an item so that other customers cannot attempt to purchase.

Integrity Testing

- Key Integrity
  - Every table must have a primary key.
- Entity Integrity
- Domain Integrity
  - Appropriate controls must be designed to ensure that no field takes on a value that is outside of the range of legal values.
  - Example, if size must be S, M, L, X2 then domain constraint wont allow data modification other than those.

Integrity testing

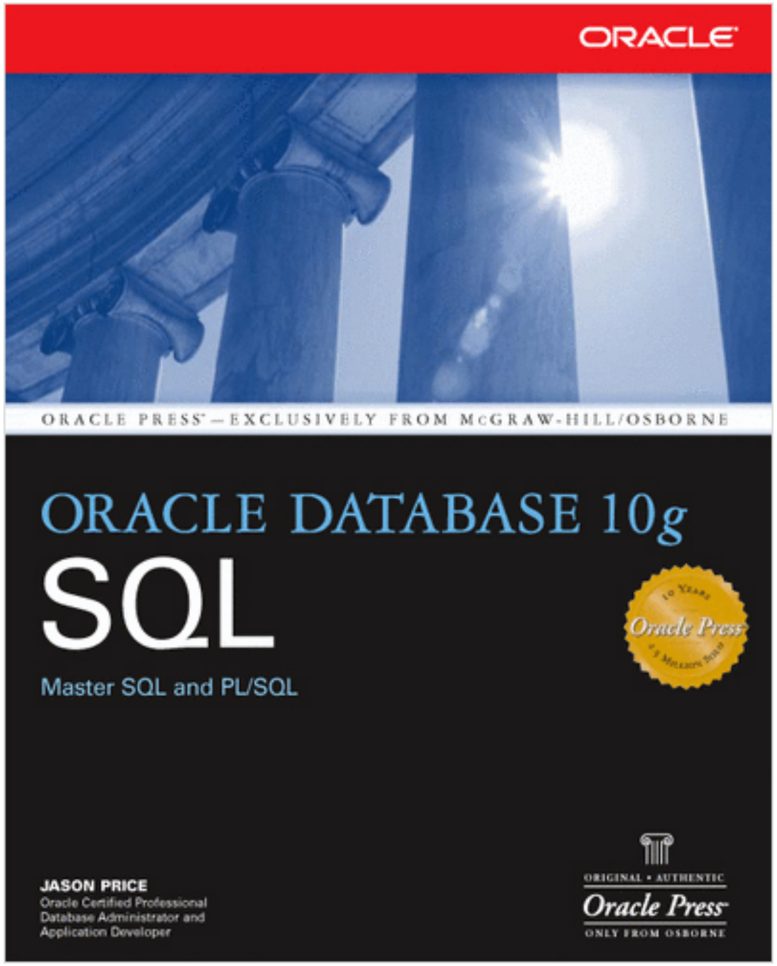
- Referential Integrity
  - One relation includes reference to another existing relation.

Preparation for Database Testing

- Can certain field value be NULL?
- What are the allowed or disallowed values?
- What are the constraints?
- Is the value dependent upon values in another table?
- Will the values of this field be in the lookup table?
- What are the user defined data types?
- What are the primary key and foreign key relationships among tables?

Book Suggestion

- Oracle Database 10g SQL
- By Jason Price
- Publisher McGraw-hill
- ISBN: 978-0072229813
- Amazon.com price : $34.45



Resources

- www.W3schools.Com
- www.Google.com