



ADVANCED DATABASE AND TECHNOLOGY

Advanced Database Management System

TOPIC: PARALLEL AND DISTRIBUTED DATABASE

UWANYIRIGIRA CLAUDINE
REG NO: 224020280 DATE: 30/10/2025

1. Introduction

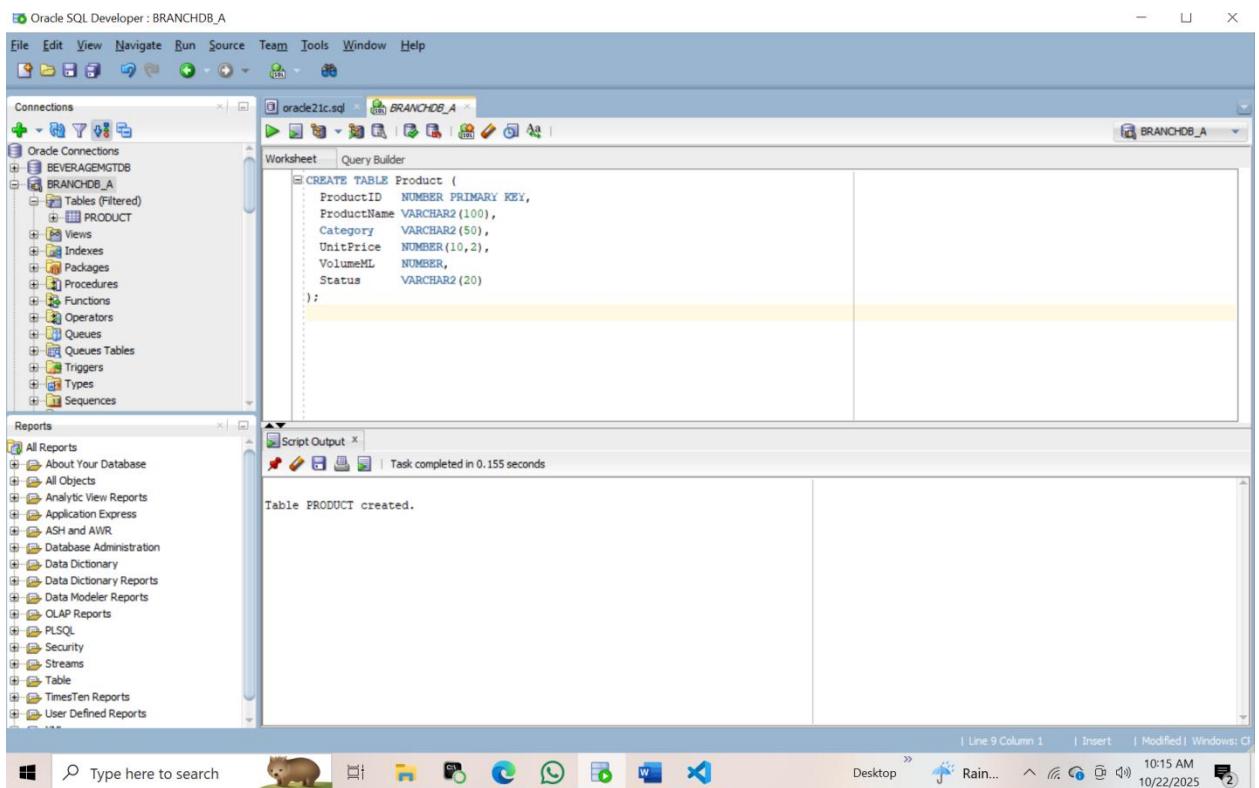
Distributed Database System Implementation Report

This report summarizes the implementation and testing phases of a distributed database system built on Oracle, focusing on data fragmentation, remote access, parallel processing, and distributed transaction management.

Q1. Distributed Schema Design and Fragmentation : we split the database into two branches using region-based horizontal fragmentation. Each branch stores its own distributors and deliveries

A. CREATING TABLES OF BRANCH_A

PRODUCT_A TABLE



The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the 'Connections' tree, which includes 'BEVERAGEGMGDB' and 'BRANCHDB_A'. Under 'BRANCHDB_A', there are 'Tables Filtered' containing a single entry 'PRODUCT'. The central workspace shows a script named 'oracle21c.sql' with the following code:

```
CREATE TABLE Product (
    ProductID NUMBER PRIMARY KEY,
    ProductName VARCHAR2(100),
    Category VARCHAR2(50),
    UnitPrice NUMBER(10,2),
    VolumeML NUMBER,
    Status VARCHAR2(20)
);
```

The 'Script Output' window at the bottom shows the message: "Table PRODUCT created." The status bar at the bottom right indicates the task completed in 0.155 seconds.

b.BATCH_A TABLE

Oracle SQL Developer : BRANCHDB_A

File Edit View Navigate Run Source Team Tools Window Help

Connections

BRANCHDB_A

Tables (Filtered)

BATCH

PRODUCT

Views

Indexes

Packages

Procedures

Functions

Operators

Queues

Queues Tables

Triggers

Types

Reports

All Reports

About Your Database

All Objects

Analytic View Reports

Application Express

ASH and AWR

Database Administration

Data Dictionary

Data Dictionary Reports

Data Modeler Reports

OLAP Reports

PLSQL

Security

Streams

Table

TimesTen Reports

User Defined Reports

oracle21c.sql

BRANCHDB_A

Worksheet Query Builder

```
CREATE TABLE Batch (
    BatchID NUMBER PRIMARY KEY,
    ProductID NUMBER,
    ProductionDate DATE,
    Quantity NUMBER,
    ExpiryDate DATE
);
```

Script Output

Task completed in 0.075 seconds

Table BATCH created.

Type here to search

Desktop 10:18 AM 10/22/2025

C.DISTRIBUTOR_A TABLE

Oracle SQL Developer : BRANCHDB_A

File Edit View Navigate Run Source Team Tools Window Help

Connections

BRANCHDB_A

Tables (Filtered)

BATCH

DISTRIBUTOR

PRODUCT

Views

Indexes

Packages

Procedures

Functions

Operators

Queues

Queues Tables

Triggers

Reports

All Reports

About Your Database

All Objects

Analytic View Reports

Application Express

ASH and AWR

Database Administration

Data Dictionary

Data Dictionary Reports

Data Modeler Reports

OLAP Reports

PLSQL

Security

Streams

Table

TimesTen Reports

User Defined Reports

oracle21c.sql

BRANCHDB_A

Worksheet Query Builder

```
CREATE TABLE Distributor (
    DistributorID NUMBER PRIMARY KEY,
    Name VARCHAR2(100),
    Contact VARCHAR2(50),
    Region VARCHAR2(50),
    LicenseNo VARCHAR2(30)
);
```

Script Output

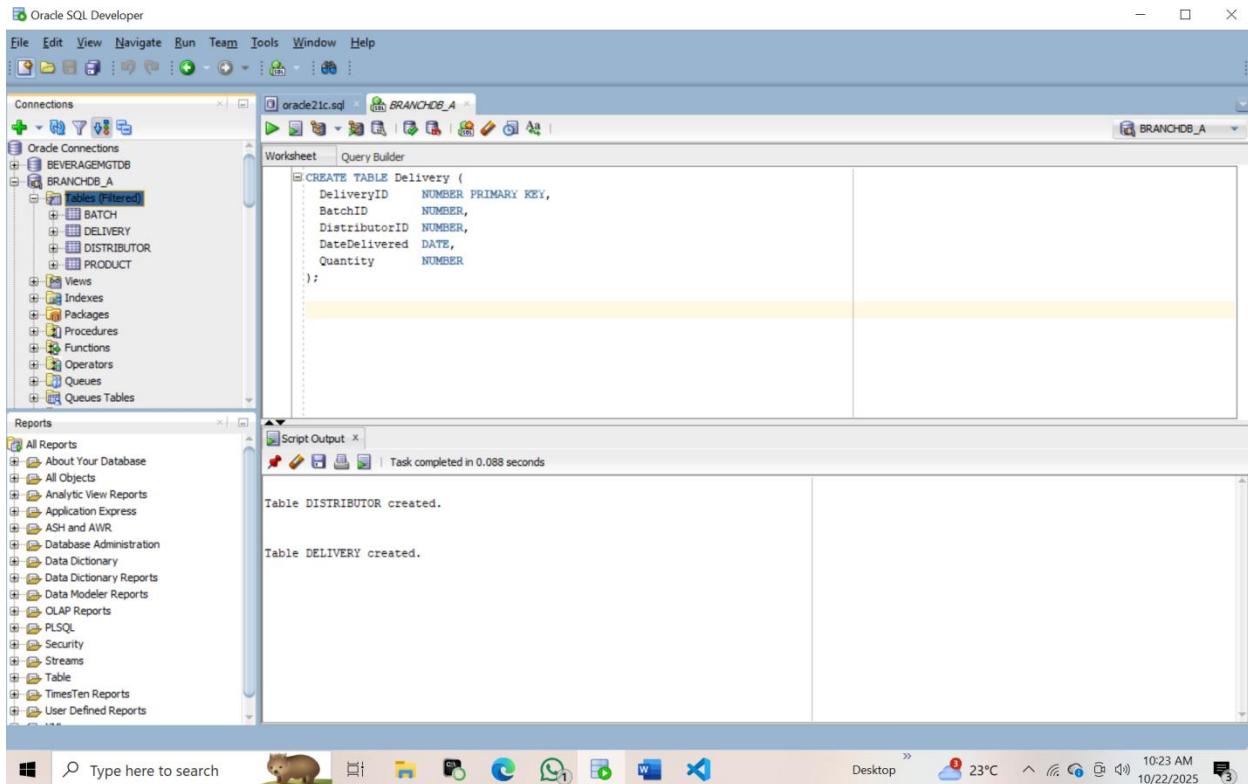
Task completed in 0.096 seconds

Table DISTRIBUTOR created.

Type here to search

Desktop 10:21 AM 10/22/2025

D.DELIVERY_A TABLE



The screenshot shows the Oracle SQL Developer interface. The 'Connections' sidebar on the left lists 'BEVERAGEMGTD8' and 'BRANCHDB_A'. Under 'Tables (Filtered)', it shows 'BATCH', 'DELIVERY', 'DISTRIBUTOR', and 'PRODUCT'. The 'Worksheet' tab in the center contains the following SQL code:

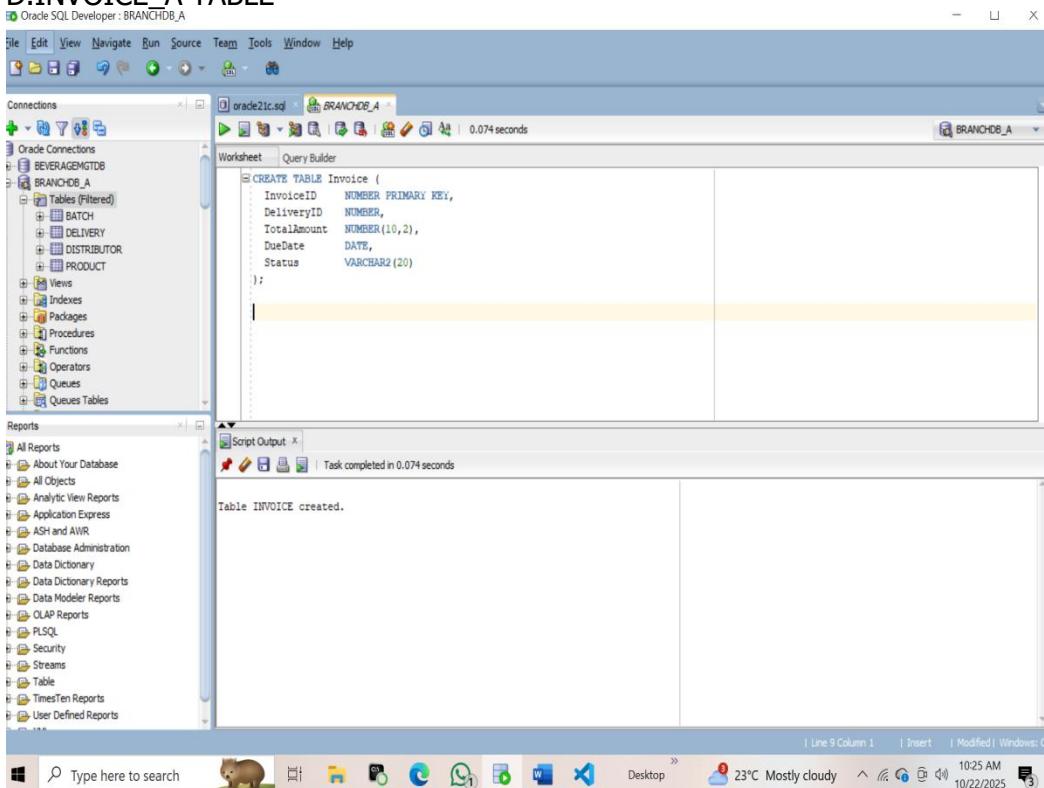
```
CREATE TABLE Delivery (
    DeliveryID NUMBER PRIMARY KEY,
    BatchID NUMBER,
    DistributorID NUMBER,
    DateDelivered DATE,
    Quantity NUMBER
);
```

The 'Script Output' pane at the bottom shows the results of the execution:

```
Table DISTRIBUTOR created.  
Table DELIVERY created.
```

The system tray at the bottom right indicates it's 10:23 AM on 10/22/2025, with a temperature of 23°C.

D.INVOICE_A TABLE



The screenshot shows the Oracle SQL Developer interface. The 'Connections' sidebar on the left lists 'BEVERAGEMGTD8' and 'BRANCHDB_A'. Under 'Tables (Filtered)', it shows 'BATCH', 'DELIVERY', 'DISTRIBUTOR', and 'PRODUCT'. The 'Worksheet' tab in the center contains the following SQL code:

```
CREATE TABLE Invoice (
    InvoiceID NUMBER PRIMARY KEY,
    DeliveryID NUMBER,
    TotalAmount NUMBER(10,2),
    DueDate DATE,
    Status VARCHAR2(20)
);
```

The 'Script Output' pane at the bottom shows the results of the execution:

```
Table INVOICE created.
```

The system tray at the bottom right indicates it's 10:25 AM on 10/22/2025, with a temperature of 23°C.

E. PAYMENT_A TABLE

The screenshot shows the Oracle SQL Developer interface for the database BRANCHDB_A. The 'Connections' sidebar on the left lists 'BRANCHDB_A' with its tables: BATCH, DELIVERY, DISTRIBUTOR, INVOICE, PAYMENT, and PRODUCT. The 'Worksheet' tab contains the SQL script for creating the 'PAYMENT' table:

```
CREATE TABLE Payment (
    PaymentID NUMBER PRIMARY KEY,
    InvoiceID NUMBER,
    Amount NUMBER(10,2),
    PaymentDate DATE,
    Method VARCHAR2(30)
);
```

The 'Script Output' pane below the worksheet shows the confirmation message: "Table PAYMENT created." The status bar at the bottom right indicates the task completed in 0.091 seconds.

B. CREATING TABLE FOR BRANCH_B PRODUCT_B

The screenshot shows the Oracle SQL Developer interface for the database BRANCHDB_B. The 'Connections' sidebar on the left lists 'BRANCHDB_B' with its tables: PRODUCT_B. The 'Worksheet' tab contains the SQL script for creating the 'Product_b' table:

```
CREATE TABLE Product_b (
    ProductID NUMBER PRIMARY KEY,
    ProductName VARCHAR2(100),
    Category VARCHAR2(50),
    UnitPrice NUMBER(10,2),
    VolumeML NUMBER,
    Status VARCHAR2(20)
);
```

The 'Script Output' pane below the worksheet shows the confirmation message: "Table PRODUCT_B created." The status bar at the bottom right indicates the task completed in 0.122 seconds.

BATCH_B

The screenshot shows the Oracle SQL Developer interface. The title bar says "BATCH_B". The menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, Help. The Connections sidebar shows a connection to "BRANCHDB_B". The Worksheet pane contains the following SQL code:

```
CREATE TABLE Batch_B (
    BatchID      NUMBER PRIMARY KEY,
    ProductID   NUMBER,
    ProductionDate DATE,
    Quantity     NUMBER,
    ExpiryDate   DATE
);
```

The Script Output pane shows the result: "Table BATCH_B created." The status bar at the bottom right shows "10:40 AM 10/22/2025".

DISTRIBUTOR_B

The screenshot shows the Oracle SQL Developer interface. The title bar says "DISTRIBUTOR_B". The menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, Help. The Connections sidebar shows a connection to "BRANCHDB_B". The Worksheet pane contains the following SQL code:

```
CREATE TABLE Distributor_B (
    DistributorID NUMBER PRIMARY KEY,
    Name          VARCHAR2(100),
    Contact       VARCHAR2(50),
    Region        VARCHAR2(50),
    LicenseNo     VARCHAR2(30)
);
```

The Script Output pane shows the result: "Table DISTRIBUTOR_B created." The status bar at the bottom right shows "10:44 AM 10/22/2025".

DELIVERY_B

BRANCHDB_B

The screenshot shows the Oracle SQL Developer interface with the title bar "Oracle SQL Developer - BRANCHDB_B". The "Connections" sidebar on the left lists "RDF Semantic Graph", "Recycle Bin", "Other Users", and "BRANCHDB_B" which is expanded to show "Tables (Filtered)" containing "BATCH_B", "DISTRIBUTOR_B", "PRODUCT_B", and "DELIVERY_B". The "Reports" sidebar lists various database reports. The main workspace has two tabs: "oracle21c.sql" and "BRANCHDB_A" (active) and "BRANCHDB_B". The "Worksheet" tab contains the SQL code:

```

CREATE TABLE Delivery_b (
    DeliveryID NUMBER PRIMARY KEY,
    BatchID NUMBER,
    DistributorID NUMBER,
    DateDelivered DATE,
    Quantity NUMBER
);

```

The "Script Output" tab shows the result: "Table DELIVERY_B created." The status bar at the bottom right indicates "Line 5 Column 23 | Insert | Modified | Windows: C | 10:46 AM | 10/22/2025".

INVOICE_B

The screenshot shows the Oracle SQL Developer interface with the title bar "Oracle SQL Developer : BRANCHDB_B". The "Connections" sidebar on the left lists "RDF Semantic Graph", "Recycle Bin", "Other Users", and "BRANCHDB_B" which is expanded to show "Tables (Filtered)" containing "BATCH_B", "DELIVERY_B", "DISTRIBUTOR_B", "INVOICE_B", and "PRODUCT_B". The "Reports" sidebar lists various database reports. The main workspace has two tabs: "oracle21c.sql" and "BRANCHDB_A" (active) and "BRANCHDB_B". The "Worksheet" tab contains the SQL code:

```

CREATE TABLE Invoice_B(
    InvoiceID NUMBER PRIMARY KEY,
    DeliveryID NUMBER,
    TotalAmount NUMBER(10,2),
    DueDate DATE,
    Status VARCHAR2(20)
);

```

The "Script Output" tab shows the result: "Table INVOICE_B created." The status bar at the bottom right indicates "Line 10 Column 1 | Insert | Modified | Windows: C | 10:49 AM | 10/22/2025".

PAYMENT_B

Oracle SQL Developer : BRANCHDB_B

File Edit View Navigate Run Source Team Tools Window Help

Connections

- RDF Semantic Graph
- Recycle Bin
- Other Users
- BRANCHDB_B**
 - Tables (Filtered)
 - BATCH_B
 - DELIVERY_B
 - DISTRIBUTOR_B
 - INVOICE_B
 - PAYMENT_B
 - PRODUCT_B
 - Views
 - Indexes
 - Packages
 - Procedures
 - Functions

Reports

- All Reports
- About Your Database
- All Objects
- Analytic View Reports
- Application Express
- ASH and AWR
- Database Administration
- Data Dictionary
- Data Dictionary Reports
- Data Modeler Reports
- OLAP Reports
- PLSQL
- Security
- Streams
- Table
- TimesTen Reports
- User Defined Reports

oracle21c.sql [BRANCHDB_A] [BRANCHDB_B]

Worksheet Query Builder

```
CREATE TABLE Payment_B (
  PaymentID NUMBER PRIMARY KEY,
  InvoiceID NUMBER,
  Amount NUMBER(10, 2),
  PaymentDate DATE,
  Method VARCHAR2(30)
);
```

Script Output X | Task completed in 0.069 seconds

Table PAYMENT_B created.

Line 9 Column 1 | Insert | Modified | Windows: C

C.INSERTING DATA IN BRANCH_A TABLES

PRODUCT_A

Oracle SQL Developer : BRANCHDB_A

File Edit View Navigate Run Source Team Tools Window Help

Connections

- Tables (Filtered)
 - BATCH
 - DELIVERY
 - DISTRIBUTOR
 - INVOICE
 - PAYMENT
 - PRODUCT
- Views
- Indexes
- Packages
- Procedures
- Functions
- Operators
- Queues
- Queues Tables
- Triggers

Reports

- All Reports
- About Your Database
- All Objects
- Analytic View Reports
- Application Express
- ASH and AWR
- Database Administration
- Data Dictionary
- Data Dictionary Reports
- Data Modeler Reports
- OLAP Reports
- PLSQL
- Security
- Streams
- Table
- TimesTen Reports
- User Defined Reports

oracle21c.sql [BRANCHDB_A] [BRANCHDB_B] [BEVERAGEGTDB]

Worksheet Query Builder

```
INSERT INTO Product VALUES (101, 'Inyangi Juice', 'Soft Drink', 500, 500, 'Active');
INSERT INTO Product VALUES (102, 'Skol Lager', 'Alcoholic', 800, 330, 'Active');
INSERT INTO Product VALUES (103, 'Mutzi Water', 'Water', 300, 1000, 'Active');
INSERT INTO Product VALUES (104, 'Primus Beer', 'Alcoholic', 1000, 500, 'Discontinued');
INSERT INTO Product VALUES (105, 'Vital Yogurt', 'Dairy', 700, 250, 'Active');
select * from product;
```

Script Output X | Query Result X | All Rows Fetched: 5 in 0.022 seconds

PRODUCTID	PRODUCTNAME	CATEGORY	UNITPRICE	VOLUMEML	STATUS
1	101 Inyangi Juice	Soft Drink	500	500	Active
2	102 Skol Lager	Alcoholic	800	330	Active
3	103 Mutzi Water	Water	300	1000	Active
4	104 Primus Beer	Alcoholic	1000	500	Discontinued
5	105 Vital Yogurt	Dairy	700	250	Active

Line 6 Column 22 | Insert | Modified | Windows: C

BATCH_A DATA

File Edit View Navigate Run Source Team Tools Window Help

Connections: BRANCHDB_A, BRANCHDB_B, BEVERAGEGTDB

Worksheet: Query Builder

```
INSERT INTO Batch VALUES (201, 101, TO_DATE('2025-10-01', 'YYYY-MM-DD'), 1000, TO_DATE('2026-04-01', 'YYYY-MM-DD'));
INSERT INTO Batch VALUES (202, 102, TO_DATE('2025-09-15', 'YYYY-MM-DD'), 800, TO_DATE('2026-03-15', 'YYYY-MM-DD'));
INSERT INTO Batch VALUES (203, 103, TO_DATE('2025-10-10', 'YYYY-MM-DD'), 1200, TO_DATE('2026-05-10', 'YYYY-MM-DD'));
INSERT INTO Batch VALUES (204, 104, TO_DATE('2025-08-20', 'YYYY-MM-DD'), 500, TO_DATE('2026-02-20', 'YYYY-MM-DD'));
INSERT INTO Batch VALUES (205, 105, TO_DATE('2025-10-05', 'YYYY-MM-DD'), 950, TO_DATE('2026-04-05', 'YYYY-MM-DD'));
select * from BATCH;
```

Script Output: All Rows Fetched: 5 in 0.014 seconds

BATCHID	PRODUCTID	PRODUCTIONDATE	QUANTITY	EXPIRYDATE
1	201	101 01-OCT-25	1000	01-APR-26
2	202	102 15-SEP-25	800	15-MAR-26
3	203	103 10-OCT-25	1200	10-MAY-26
4	204	104 20-AUG-25	500	20-FEB-26
5	205	105 05-OCT-25	950	05-APR-26

Reports: All Reports, About Your Database, All Objects, Analytic View Reports, Application Express, ASH and AWR, Database Administration, Data Dictionary, Data Dictionary Reports, Data Modeler Reports, OLAP Reports, PLSQL, Security, Streams, Table, TimesTen Reports, User Defined Reports

Type here to search: Desktop 24°C 11:39 AM 10/22/2025

Distributor_a data

File Edit View Navigate Run Source Team Tools Window Help

Connections: BRANCHDB_A, BRANCHDB_B, BEVERAGEGTDB

Worksheet: Query Builder

```
INSERT INTO Distributor VALUES (301, 'Kigali Beverages Ltd', '0788123456', 'Kigali', 'LIC-KGL-001');
INSERT INTO Distributor VALUES (302, 'Musanzé Drinks Co', '0788234567', 'Musanzé', 'LIC-MUS-002');
INSERT INTO Distributor VALUES (303, 'Rubavu Refreshments', '0788345678', 'Rubavu', 'LIC-RBV-003');
INSERT INTO Distributor VALUES (304, 'Huye Hydration Hub', '0788456789', 'Huye', 'LIC-HUY-004');
INSERT INTO Distributor VALUES (305, 'Rwamagana Bottlers', '0788567890', 'Rwamagana', 'LIC-RWA-005');
SELECT * FROM Distributor;
```

Script Output: All Rows Fetched: 5 in 0.009 seconds

DISTRIBUTORID	NAME	CONTACT	REGION	LICENSENO
1	301 Kigali Beverages Ltd	0788123456	Kigali	LIC-KGL-001
2	302 Musanzé Drinks Co	0788234567	Musanzé	LIC-MUS-002
3	303 Rubavu Refreshments	0788345678	Rubavu	LIC-RBV-003
4	304 Huye Hydration Hub	0788456789	Huye	LIC-HUY-004
5	305 Rwamagana Bottlers	0788567890	Rwamagana	LIC-RWA-005

Reports: All Reports, About Your Database, All Objects, Analytic View Reports, Application Express, ASH and AWR, Database Administration, Data Dictionary, Data Dictionary Reports, Data Modeler Reports, OLAP Reports, PLSQL, Security, Streams, Table, TimesTen Reports, User Defined Reports

Type here to search: Desktop 24°C Mostly cloudy 11:44 AM 10/22/2025

Delivery_A data

Oracle SQL Developer : BRANCHDB_A

File Edit View Navigate Run Source Team Tools Window Help

Connections

- Tables (Filtered)
 - BATCH
 - DELIVERY
 - DISTRIBUTOR
 - INVOICE
 - PAYMENT
 - PRODUCT
- Views
- Indexes
- Packages
- Procedures
- Functions
- Operators
- Queues
- Queues Tables
- Triggers

Reports

- All Reports
- About Your Database
- All Objects
- Analytic View Reports
- Application Express
- ASH and AWR
- Database Administration
- Data Dictionary
- Data Dictionary Reports
- Data Modeler Reports
- OLAP Reports
- PLSQL
- Security
- Streams
- Table
- TimesTen Reports
- User Defined Reports

Worksheet Query Builder

```
orade21c.sql BRANCHDB_A BRANCHDB_B BEVERAGEGTDDB
```

INSERT INTO Delivery VALUES (401, 201, 301, TO_DATE('2025-10-15', 'YYYY-MM-DD'), 500);
INSERT INTO Delivery VALUES (402, 202, 302, TO_DATE('2025-10-16', 'YYYY-MM-DD'), 400);
INSERT INTO Delivery VALUES (403, 203, 303, TO_DATE('2025-10-17', 'YYYY-MM-DD'), 600);
INSERT INTO Delivery VALUES (404, 204, 304, TO_DATE('2025-10-18', 'YYYY-MM-DD'), 300);
INSERT INTO Delivery VALUES (405, 205, 305, TO_DATE('2025-10-19', 'YYYY-MM-DD'), 450);
select * from delivery;

Script Output | Query Result | Query Result 1

All Rows Fetched: 5 in 0.01 seconds

DELIVERYID	BATCHID	DISTRIBUTORID	DATEDELIVERED	QUANTITY
1	401	201	301 15-OCT-25	500
2	402	202	302 16-OCT-25	400
3	403	203	303 17-OCT-25	600
4	404	204	304 18-OCT-25	300
5	405	205	305 19-OCT-25	450

Line 8 Column 1 | Insert | Modified | Windows: C

Type here to search

Windows Taskbar: Desktop, Weather, Home, Microsoft Edge, File Explorer, Task View, Word, Excel, Powerpoint, File History, Taskbar Icons, 11:50 AM, 10/22/2025

Invoice_A

File Edit View Navigate Run Source Team Tools Window Help

Connections

- Tables (Filtered)
 - BATCH
 - DELIVERY
 - DISTRIBUTOR
 - INVOICE
 - PAYMENT
 - PRODUCT
- Views
- Indexes
- Packages
- Procedures
- Functions
- Operators
- Queues
- Queues Tables
- Triggers

Reports

- All Reports
- About Your Database
- All Objects
- Analytic View Reports
- Application Express
- ASH and AWR
- Database Administration
- Data Dictionary
- Data Dictionary Reports
- Data Modeler Reports
- OLAP Reports
- PLSQL
- Security
- Streams
- Table
- TimesTen Reports
- User Defined Reports

Worksheet Query Builder

```
orade21c.sql BRANCHDB_A BRANCHDB_B BEVERAGEGTDDB
```

INSERT INTO Invoice VALUES (501, 401, 250000, TO_DATE('2025-11-15', 'YYYY-MM-DD'), 'Unpaid');
INSERT INTO Invoice VALUES (502, 402, 180000, TO_DATE('2025-11-16', 'YYYY-MM-DD'), 'Paid');
INSERT INTO Invoice VALUES (503, 403, 300000, TO_DATE('2025-11-17', 'YYYY-MM-DD'), 'Unpaid');
INSERT INTO Invoice VALUES (504, 404, 120000, TO_DATE('2025-11-18', 'YYYY-MM-DD'), 'Paid');
INSERT INTO Invoice VALUES (505, 405, 225000, TO_DATE('2025-11-19', 'YYYY-MM-DD'), 'Overdue');
SELECT * FROM invoice;

Script Output | Query Result | Query Result 1 | Query Result 2

All Rows Fetched: 5 in 0.01 seconds

INVOICEID	DELIVERYID	TOTALAMOUNT	DUEDATE	STATUS
1	501	401	250000 15-NOV-25	Unpaid
2	502	402	180000 16-NOV-25	Paid
3	503	403	300000 17-NOV-25	Unpaid
4	504	404	120000 18-NOV-25	Paid
5	505	405	225000 19-NOV-25	Overdue

Line 7 Column 1 | Insert | Modified | Windows: C

Type here to search

Windows Taskbar: Desktop, Weather, Home, Microsoft Edge, File Explorer, Task View, Word, Excel, Powerpoint, File History, Taskbar Icons, 11:54 AM, 10/22/2025

Payment_A data

Oracle SQL Developer : BRANCHDB_A

File Edit View Navigate Run Source Team Tools Window Help

Connections

Worksheet Query Builder

```
INSERT INTO Payment VALUES (601, 501, 250000, TO_DATE('2025-11-10', 'YYYY-MM-DD'), 'Mobile Money');
INSERT INTO Payment VALUES (602, 502, 180000, TO_DATE('2025-11-11', 'YYYY-MM-DD'), 'Bank Transfer');
INSERT INTO Payment VALUES (603, 503, 300000, TO_DATE('2025-11-12', 'YYYY-MM-DD'), 'Cash');
INSERT INTO Payment VALUES (604, 504, 120000, TO_DATE('2025-11-13', 'YYYY-MM-DD'), 'Mobile Money');
INSERT INTO Payment VALUES (605, 505, 225000, TO_DATE('2025-11-14', 'YYYY-MM-DD'), 'Bank Transfer');
select * from payment;
```

Script Output | Query Result | Query Result 1 | Query Result 2 | Query Result 3 | Query Result 4

All Reports | About Your Database | All Objects | Analytic View Reports | Application Express | ASH and AWR | Database Administration | Data Dictionary | Data Dictionary Reports | Data Modeler Reports | OLAP Reports | PLSQL | Security | Streams | Table | TimesTen Reports | User Defined Reports

Reports

Type here to search

Line 8 Column 1 | Insert | Modified | Windows: C

Desktop 24°C 12:01 PM 10/22/2025

D.INSERTING DATA IN BRANCH_B TABLES

Product_b

Oracle SQL Developer : BRANCHDB_B

File Edit View Navigate Run Source Team Tools Window Help

Connections

Worksheet Query Builder

```
INSERT INTO product_b VALUES (107, 'Inyange Juice', 'Soft Drink', 500.00, 500, 'Active');
INSERT INTO product_b VALUES (108, 'Skol Lager', 'Alcoholic', 800.00, 330, 'Active');
INSERT INTO product_b VALUES (109, 'Mutti Water', 'Water', 300.00, 1000, 'Active');
INSERT INTO Product_b VALUES (110, 'Primus Beer', 'Alcoholic', 1000.00, 500, 'Discontinued');
INSERT INTO product_b VALUES (111, 'Vital Yogurt', 'Dairy', 700.00, 250, 'Active');

select * from product_b
```

Script Output | Query Result

All Reports | About Your Database | All Objects | Analytic View Reports | Application Express | ASH and AWR | Database Administration | Data Dictionary | Data Dictionary Reports | Data Modeler Reports | OLAP Reports | PLSQL | Security | Streams | Table | TimesTen Reports | User Defined Reports

Reports

Type here to search

Line 11 Column 1 | Insert | Modified | Windows: C

Desktop 24°C Mostly cloudy 12:07 PM 10/22/2025

Batch_b data

Oracle SQL Developer: BRANCHDB_B

The screenshot shows the Oracle SQL Developer interface with the connection 'BRANCHDB_B' selected. In the 'Worksheet' tab, a script is run to insert four rows of data into the 'BATCH_B' table. The data is as follows:

```

INSERT INTO BATCH_B (BatchID, ProductID, ProductionDate, Quantity, ExpiryDate)
VALUES (111, 107, TO_DATE('2025-10-01', 'YYYY-MM-DD'), 500, TO_DATE('2026-04-01', 'YYYY-MM-DD'));

INSERT INTO BATCH_B (BatchID, ProductID, ProductionDate, Quantity, ExpiryDate)
VALUES (112, 108, TO_DATE('2025-10-10', 'YYYY-MM-DD'), 800, TO_DATE('2026-01-10', 'YYYY-MM-DD'));

INSERT INTO BATCH_B (BatchID, ProductID, ProductionDate, Quantity, ExpiryDate)
VALUES (113, 109, TO_DATE('2025-09-20', 'YYYY-MM-DD'), 300, TO_DATE('2026-03-20', 'YYYY-MM-DD'));

INSERT INTO BATCH_B (BatchID, ProductID, ProductionDate, Quantity, ExpiryDate)
VALUES (114, 110, TO_DATE('2025-08-15', 'YYYY-MM-DD'), 1000, TO_DATE('2025-12-15', 'YYYY-MM-DD'));
select * from BATCH_B;

```

The 'Query Result 1' tab shows the inserted data:

BATCHID	PRODUCTID	PRODUCTIONDATE	QUANTITY	EXPIRYDATE
1	111	107 01-OCT-25	500	01-APR-26
2	112	108 10-OCT-25	800	10-JAN-26
3	113	109 20-SEP-25	300	20-MAR-26
4	114	110 15-AUG-25	1000	15-DEC-25

Distributor_b data

Oracle SQL Developer: BRANCHDB_B

The screenshot shows the Oracle SQL Developer interface with the connection 'BRANCHDB_B' selected. In the 'Worksheet' tab, a script is run to insert three rows of data into the 'distributor_b' table. The data is as follows:

```

INSERT INTO distributor_b (DistributorID, Name, Contact, Region, LicenseNo)
VALUES (201, 'Inyangi Distributors Ltd', '0788123456', 'Kigali', 'LIC-RW-001');

INSERT INTO distributor_b (DistributorID, Name, Contact, Region, LicenseNo)
VALUES (202, 'Skol Distribution Co.', '0788456789', 'Northern', 'LIC-RW-002');

INSERT INTO distributor_b (DistributorID, Name, Contact, Region, LicenseNo)
VALUES (203, 'Mutzi Supply Chain', '0788567890', 'Eastern', 'LIC-RW-003');

select * from Distributor_b;

```

The 'Query Result' tab shows the inserted data:

DISTRIBUTORID	NAME	CONTACT	REGION	LICENSENO
1	201 Inyangi Distributors Ltd	0788123456	Kigali	LIC-RW-001
2	202 Skol Distribution Co.	0788456789	Northern	LIC-RW-002
3	203 Mutzi Supply Chain	0788567890	Eastern	LIC-RW-003

Delivery_b data

Oracle SQL Developer : BRANCHDB_B

File Edit View Navigate Run Source Team Tools Window Help

Connections

- Oracle Connections
 - BEVERAGEGMGDB
 - BRANCHDB_A
 - BRANCHDB_B
 - Tables (Filtered)
 - BATCH_B
 - DELIVERY_B
 - DISTRIBUTOR_B
 - INVOICE_B
 - PAYMENT_B
 - PRODUCT_B
 - Views
 - Indexes
 - Packages
 - Procedures
 - Functions

Worksheet Query Builder

```
INSERT INTO delivery_b (DeliveryID, BatchID, DistributorID, DateDelivered, Quantity)
VALUES (401, 111, 201, TO_DATE('2025-10-05', 'YYYY-MM-DD'), 200);

INSERT INTO Delivery_b(DeliveryID, BatchID, DistributorID, DateDelivered, Quantity)
VALUES (402, 112, 202, TO_DATE('2025-10-12', 'YYYY-MM-DD'), 150);

INSERT INTO delivery_b (DeliveryID, BatchID, DistributorID, DateDelivered, Quantity)
VALUES (403, 113, 203, TO_DATE('2025-09-25', 'YYYY-MM-DD'), 100);

SELECT * FROM delivery_b;
```

Script Output | Query Result | Query Result 1 | Query Result 2 | Query Result 3 |

DELIVERYID	BATCHID	DISTRIBUTORID	DATEDELIVERED	QUANTITY
1	401	111	201 05-OCT-25	200
2	402	112	202 12-OCT-25	150
3	403	113	203 25-SEP-25	100

All Reports

- About Your Database
- All Objects
- Analytic View Reports
- Application Express
- ASH and AWR
- Database Administration
- Data Dictionary
- Data Dictionary Reports
- Data Modeler Reports
- OLAP Reports
- PLSQL
- Security
- Streams
- Table
- TimesTen Reports
- User Defined Reports

Type here to search

Line 12 Column 1 Insert Modified Windows: C

10:58 AM 10/24/2025

Invoice_b data

Oracle SQL Developer : BRANCHDB_B

File Edit View Navigate Run Source Team Tools Window Help

Connections

- Oracle Connections
 - BEVERAGEGMGDB
 - BRANCHDB_A
 - BRANCHDB_B
 - Tables (Filtered)
 - BATCH_B
 - DELIVERY_B
 - DISTRIBUTOR_B
 - INVOICE_B
 - PAYMENT_B
 - PRODUCT_B
 - Views
 - Indexes
 - Packages
 - Procedures
 - Functions

Worksheet Query Builder

```
INSERT INTO Invoice_b (InvoiceID, DeliveryID, TotalAmount, DueDate, Status)
VALUES (601, 401, 120000.00, TO_DATE('2025-11-05', 'YYYY-MM-DD'), 'Pending');

INSERT INTO Invoice_b (InvoiceID, DeliveryID, TotalAmount, DueDate, Status)
VALUES (602, 402, 95000.00, TO_DATE('2025-11-12', 'YYYY-MM-DD'), 'Paid');

INSERT INTO Invoice_b (InvoiceID, DeliveryID, TotalAmount, DueDate, Status)
VALUES (603, 403, 75000.00, TO_DATE('2025-10-30', 'YYYY-MM-DD'), 'Overdue');

SELECT * FROM invoice_b;
```

Script Output | Query Result | Query Result 1 | Query Result 2 | Query Result 3 | Query Result 4 | Query Result 5 | Query Result 6 |

INVOICEID	DELIVERYID	TOTALAMOUNT	DUEDATE	STATUS
1	601	401	120000 05-NOV-25	Pending
2	602	402	95000 12-NOV-25	Paid
3	603	403	75000 30-OCT-25	Overdue

All Reports

- About Your Database
- All Objects
- Analytic View Reports
- Application Express
- ASH and AWR
- Database Administration
- Data Dictionary
- Data Dictionary Reports
- Data Modeler Reports
- OLAP Reports
- PLSQL
- Security
- Streams
- Table
- TimesTen Reports
- User Defined Reports

Type here to search

Line 10 Column 25 Insert Modified Windows: C

11:13 AM 10/24/2025

Payment_b data

```

INSERT INTO payment_b (PaymentID, InvoiceID, Amount, PaymentDate, Method)
VALUES (701, 601, 120000.00, TO_DATE('2025-11-06', 'YYYY-MM-DD'), 'Mobile Money');

INSERT INTO Payment_b (PaymentID, InvoiceID, Amount, PaymentDate, Method)
VALUES (702, 602, 95000.00, TO_DATE('2025-11-13', 'YYYY-MM-DD'), 'Bank Transfer');

INSERT INTO payment_b (PaymentID, InvoiceID, Amount, PaymentDate, Method)
VALUES (703, 603, 75000.00, TO_DATE('2025-10-31', 'YYYY-MM-DD'), 'Cash');

COMMIT;
SELECT * FROM Payment_b;

```

PAYMENTID	INVOICEID	AMOUNT	PAYMENTDATE	METHOD
1	701	601	120000 06-NOV-25	Mobile Money
2	702	602	95000 13-NOV-25	Bank Transfer
3	703	603	75000 31-OCT-25	Cash

Q2.Create and Use Database Links:A database link allows remote queries and joins between branches.

CONNECT AS BRANCH_A

Oracle SQL Developer : BRANCHDB_A~1

File Edit View Navigate Run Source Team Tools Window Help

Connections

- + BEVERAGEMGTDB
- + BRANCHDB_A
- + BRANCHDB_B
- + oracle21c
- + system
- Database Schema Service Conn

Worksheet

```
CREATE DATABASE LINK branch_b_link
CONNECT TO branch_b IDENTIFIED BY "12345"
USING '(DESCRIPTION=
(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(PORT=1521))
(CONNECT_DATA=(SERVICE_NAME=XEPDB1))
);'
```

Script Output

Task completed in 0.04 seconds

Database link BRANCH_B_LINK created.

Reports

- All Reports
- About Your Database
- All Objects
- Analytic View Reports
- Application Express
- ASH and AWR
- Database Administration
- Data Dictionary
- Data Dictionary Reports
- Data Modeler Reports
- OLAP Reports
- PLSQL
- Security
- Streams
- Table
- TimesTen Reports
- User Defined Reports

Type here to search

Desktop 24°C Mostly cloudy 6:25 PM 10/24/2025

REMOTE Oracle SQL Developer : BRANCHDB_A

File Edit View Navigate Run Source Team Tools Window Help

Connections

- + BEVERAGEMGTDB
- + BRANCHDB_A
- + BRANCHDB_B
- + oracle21c
- + system
- Database Schema Service Conn

Worksheet

```
SELECT * FROM PRODUCT_B@branch_b_link;
SELECT * FROM BATCH_B @branch_b_link;
SELECT ProductID, ProductName, UnitPrice
FROM Product_B@branch_b_link
WHERE UnitPrice > 100;
```

Script Output

All Rows Fetched: 4 in 0.057 seconds

PRODUCTID	PRODUCTNAME	UNITPRICE
1	Inyangi Juice	500
2	Skol Lager	800
3	Mutzi Water	300
4	Primus Beer	1000

Reports

- All Reports
- About Your Database
- All Objects
- Analytic View Reports
- Application Express
- ASH and AWR
- Database Administration
- Data Dictionary
- Data Dictionary Reports
- Data Modeler Reports
- OLAP Reports
- PLSQL
- Security
- Streams
- Table
- TimesTen Reports
- User Defined Reports

77°F Mostly sunny

Search

4:29 PM 10/26/2025

--Distributed Join Between Local and Remote Tables

Oracle SQL Developer : BRANCHDB_A

File Edit View Navigate Run Source Team Tools Window Help

Connections BEVERAGEGTDB BRANCHDB_A BEVERAGEGTDB BRANCHDB_B

Worksheet Query Builder

```
SELECT InvoiceID,ProductID,ProductName, UnitPrice
FROM Invoice
JOIN Product_B@branch_b_link ON ProductID = ProductID
WHERE UnitPrice > 100;
```

Reports All Reports About Your Database All Objects Analytic View Reports Application Express ASH and AWR Database Administrator Data Dictionary Reports Data Modeler Reports OLAP Reports PLSQL Security Streams Table TimesTen Reports User Defined Procedures

Script Output x Query Result x

All Rows Fetched: 20 in 0.043 seconds

INVOICEID	PRODUCTID	PRODUCTNAME	UNITPRICE
1	501	107Inyange Juice	500
2	502	107Inyange Juice	500
3	503	107Inyange Juice	500
4	504	107Inyange Juice	500
5	505	107Inyange Juice	500
6	501	108 Skol Lager	800
7	502	108 Skol Lager	800
8	503	108 Skol Lager	800
9	504	108 Skol Lager	800
10	505	108 Skol Lager	800
11	501	109Mutzi Water	300
12	502	109Mutzi Water	300

Line 5 Column 1 Insert Modified Windows: C 4:42 PM 10/26/2025

Q3.Parallel Query Execution:

Parallel query execution helps speed up large queries by using multiple CPU threads. You can compare performance using EXPLAIN PLAN and see how much faster it runs.

Enable Parallelism on the Table

Oracle SQL Developer : BRANCHDB_B

File Edit View Navigate Run Source Team Tools Window Help

Connections: BEVERAGEMGTDDB, BRANCHDB_A, BEVERAGEMGTDDB, BRANCHDB_B

Worksheet: ALTER TABLE PRODUCT_B PARALLEL 8;

Script Output: Table PRODUCT_B altered.

Query Result: Task completed in 0.061 seconds

Reports: All Reports, About Your Database, All Objects, Analytic View Reports, Application Express, ASH and AWR, Database Administrator, Data Dictionary, Data Dictionary Reports, Data Modeler Reports, OLAP Reports, PLSQL, Security, Streams, Table, TimesTen Reports, User Defined Reports

System tray: Sports headline, World Series: Va..., 5:12 PM, 10/26/2025

Serial Query (No Hint)

Oracle SQL Developer : BRANCHDB_B

File Edit View Navigate Run Source Team Tools Window Help

Connections: BEVERAGEMGTDDB, BRANCHDB_A, BEVERAGEMGTDDB, BRANCHDB_B

Worksheet: EXPLAIN PLAN FOR SELECT COUNT(*) FROM PRODUCT_B WHERE UNITPRICE > 1000;

Script Output: Task completed in 0.091 seconds

Query Result: Explained.

Reports: All Reports, About Your Database, All Objects, Analytic View Reports, Application Express, ASH and AWR, Database Administrator, Data Dictionary, Data Dictionary Reports, Data Modeler Reports, OLAP Reports, PLSQL, Security, Streams, Table, TimesTen Reports, User Defined Reports

System tray: Humid Now, 5:41 PM, 10/26/2025

TIMING SET(No Hint)

Oracle SQL Developer : BRANCHDB_B

File Edit View Navigate Run Source Team Tools Window Help

Connections

- + Oracle Connections
 - BEVERAGEGMGTD8
 - BRANCHDB_A
 - BRANCHDB_B**
 - oracle21c
 - system
 - Database Schema Service Connectic

Worksheet Query Builder

```
EXPLAIN PLAN FOR
SELECT COUNT(*) FROM PRODUCT_B WHERE UNITPRICE > 1000;
SET TIMING ON
SELECT COUNT(*) FROM PRODUCT_B WHERE UNITPRICE > 10000;
SET TIMING OFF
```

Script Output x | Query Result x | Query Result 1 x | Query Result 2 x | Query Result 3 x | Query Result 4 x

Task completed in 0.202 seconds

Explained.

>>Query Run In:Query Result 1
Elapsed: 00:00:00.083

Explained.

>>Query Run In:Query Result 2
Elapsed: 00:00:00.370

Explained.

>>Query Run In:Query Result 3
Elapsed: 00:00:00.097

(1 more...) | Line 9 Column 1 | Insert | Modified | Windows: Q

77°F Partly sunny

Search

5:47 PM 10/26/2025

Parallel Query (With Hint)

Oracle SQL Developer : BRANCHDB_B

File Edit View Navigate Run Source Team Tools Window Help

Connections

- + Oracle Connections
 - BEVERAGEGMGTD8
 - BRANCHDB_A
 - BRANCHDB_B**
 - oracle21c
 - system
 - Database Schema Service Connectic

Worksheet Query Builder

```
SET TIMING ON
SELECT /*+ PARALLEL(PRODUCT_B, 8) */ COUNT(*) FROM PRODUCT_B WHERE UNITPRICE > 10000;
SET TIMING OFF
```

Script Output x | Task completed in 0.198 seconds

Explained.

>>Query Run In:Query Result 5

Explained.

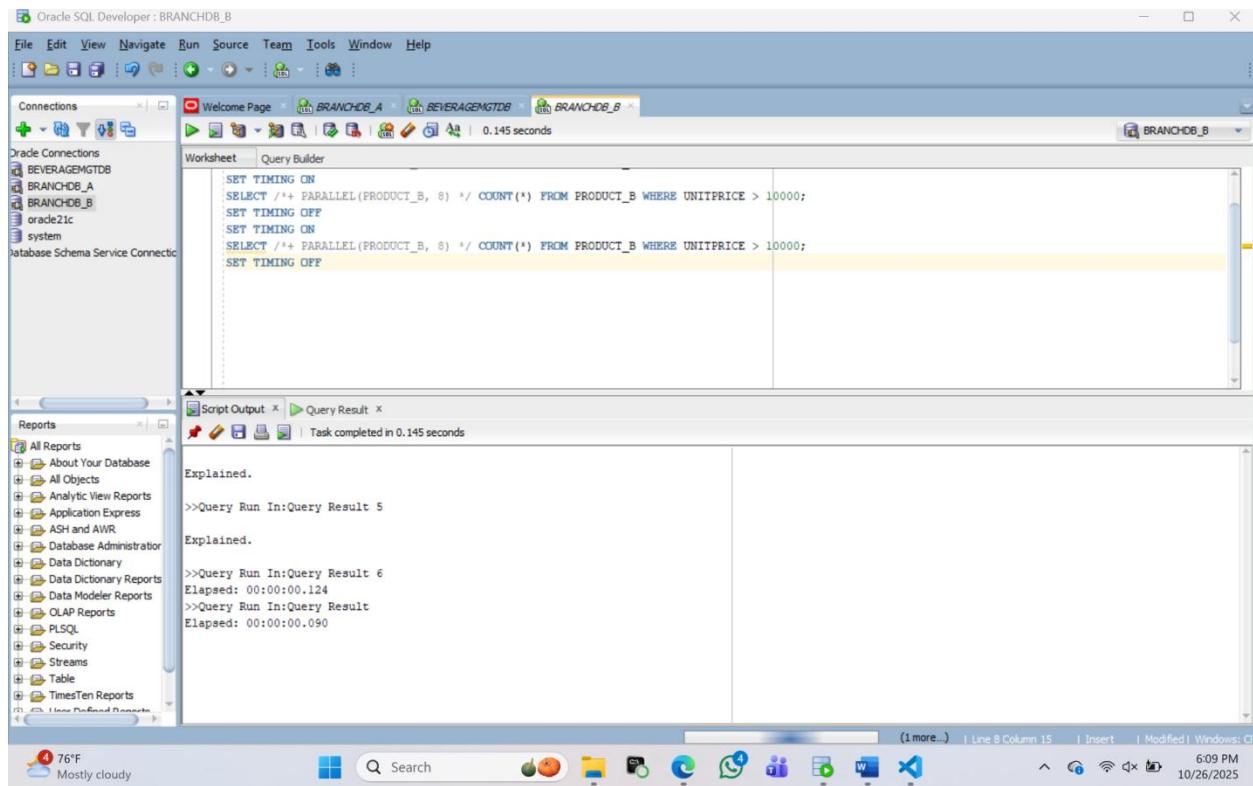
>>Query Run In:Query Result 6
Elapsed: 00:00:00.124

(1 more...) | Line 6 Column 1 | Insert | Modified | Windows: Q

76°F Mostly cloudy

Search

6:06 PM 10/26/2025



Q4.Two-Phase Commit Simulation (2PC)

I will simulate the **Two-Phase Commit (2PC)** protocol, ensuring a transaction that spans both nodes maintains atomicity.

PL/SQL Block for Distributed Insert

```

VALUES (1001, 401, 15000.00, TO_DATE('25/12/2025', 'DD/MM/YYYY'), 'PENDING');

-- Insert into remote table (Product in branch_b via DB link)
INSERT INTO Product_b@branch_b_link (ProductID, ProductName, Category, UnitPrice, VolumeML, Status)
VALUES (501, 'Mineral Water', 'Beverage', 500.00, 1000, 'AVAILABLE');

-- Commit once -- triggers Oracle's two-phase commit
COMMIT;

DBMS_OUTPUT.PUT_LINE('Both inserts committed successfully.');

EXCEPTION
  WHEN OTHERS THEN
    ROLLBACK;
    DBMS_OUTPUT.PUT_LINE('X Error: ' || SQLERRM);
END;

```

Script Output x | Task completed in 0.337 seconds

PL/SQL procedure successfully completed.

**Verify atomicity:using
DBA_2PC_PENDING.**

```

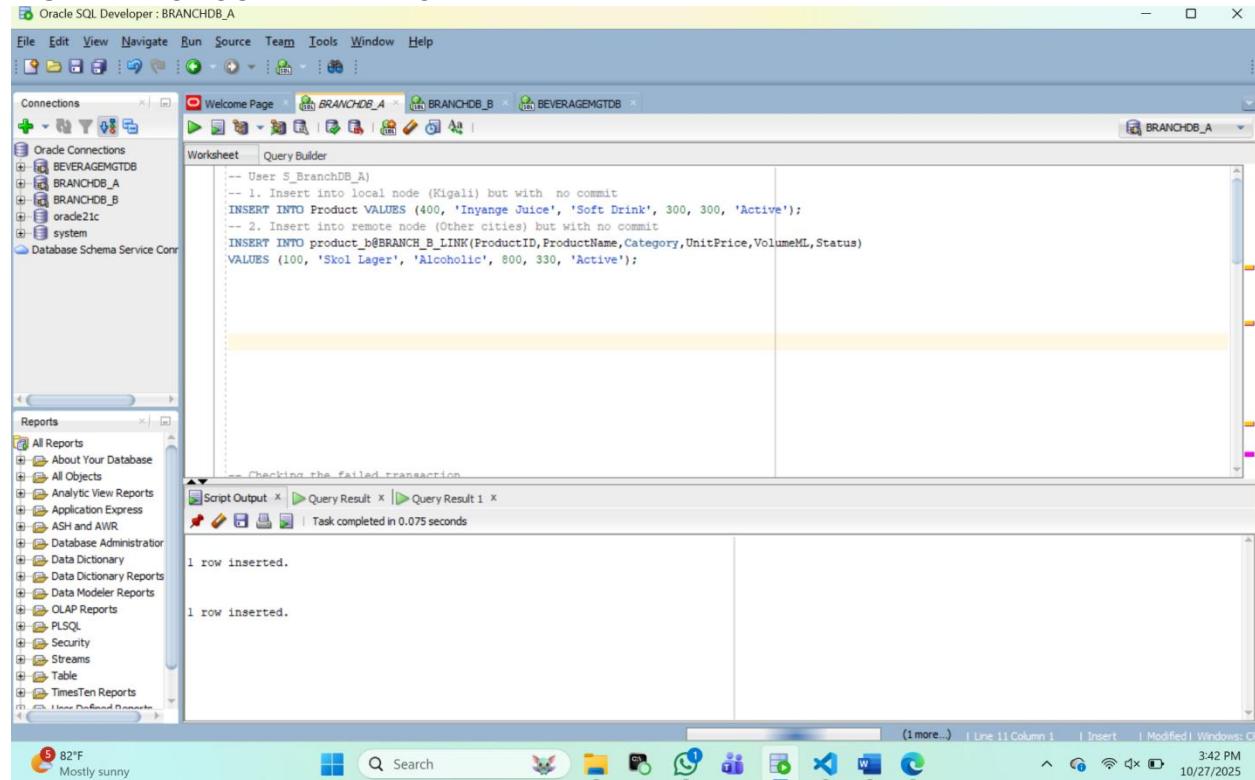
SELECT local_tran_id, global_tran_id, state, mixed, advice
FROM dba_2pc_pending;

```

LOCAL_T...	GLOBAL_T...	STATE	MIXED	ADVICE

Q5. Distributed Rollback and Recovery: I will simulate an "in-doubt" transaction resulting from a failure (e.g., network outage before commit completion) and demonstrate the recovery procedure.

Begin a distributed transaction
INSERT INTO LOCAL AND REMOTE



The screenshot shows the Oracle SQL Developer interface. The title bar says "Oracle SQL Developer : BRANCHDB_A". The menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, Help. The toolbar has various icons for connection management, navigation, and query execution. The Connections sidebar lists BEVERAGEGTDB, BRANCHDB_A, BRANCHDB_B, oracle2ic, and system. The Database Schema Service Conn section is collapsed. The Reports sidebar lists All Reports, About Your Database, All Objects, Analytic View Reports, Application Express, ASH and AWR, Database Administration, Data Dictionary, Data Dictionary Reports, Data Modeler Reports, OLAP Reports, PLSQL, Security, Streams, Table, and TimesTen Reports. The main area has tabs for Worksheet and Query Builder. The Worksheet tab contains a script with two parts: 1. Insert into local node (Kigali) but with no commit, and 2. Insert into remote node (Other cities) but with no commit. The Query Result tab shows the output: "1 row inserted." and "1 row inserted.". The status bar at the bottom right shows the date and time: 10/27/2025, 3:42 PM.

```
-- User_S_BranchDB_A]
-- 1. Insert into local node (Kigali) but with no commit
INSERT INTO Product VALUES (400, 'Inyanje Juice', 'Soft Drink', 300, 300, 'Active');

-- 2. Insert into remote node (Other cities) but with no commit
INSERT INTO product_b@BRANCH_B_LINK(ProductID,ProductName,Category,UnitPrice,VolumeML,Status)
VALUES (100, 'Skol Lager', 'Alcoholic', 800, 330, 'Active');
```

Distributed Transaction



The screenshot shows the Oracle SQL Developer interface with a "Distributed Transaction" title. The Worksheet tab contains a script with three parts: 1. SET TRANSACTION USE ROLLBACK SEGMENT rbs1; 2. INSERT INTO BRANCH_A VALUES (112, 'Branch'); 3. INSERT INTO BRANCH_B@BranchB_Link VALUES (456, 'Branch'); The Script Output tab shows the output: "transaction USE succeeded." The status bar at the bottom right shows the date and time: 10/27/2025, 3:42 PM.

```
SET TRANSACTION USE ROLLBACK SEGMENT rbs1;

INSERT INTO BRANCH_A VALUES (112, 'Branch');

INSERT INTO BRANCH_B@BranchB_Link VALUES (456, 'Branch');
```

transaction USE succeeded.

Checking the failed transaction

Oracle SQL Developer : BRANCHDB_A

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page BRANCHDB_A BRANCHDB_B BEVERAGEGTDB

Worksheet Query Builder

```
-- Checking the failed transaction
SELECT
    LOCAL_TRAN_ID,
    GLOBAL_TRAN_ID,
    STATE,
    FAIL_TIME
FROM
    DBA_2PC_PENDING
WHERE
    LOCAL_TRAN_ID IS NOT NULL
    AND LOCAL_TRAN_ID IN (SELECT DISTINCT LOCAL_TRAN_ID FROM DBA_2PC_PENDING);
```

Script Output x Query Result x

SQL | All Rows Fetched: 0 in 0.003 seconds

LOCAL_T...	GLOBAL_T...	STATE	FAIL_TIME
------------	-------------	-------	-----------

82°F Mostly sunny 3:46 PM 10/27/2025

roll back to resolve the unresolved transaction

Oracle SQL Developer : BRANCHDB_A

File Edit View Navigate Run Source Team Tools Window Help

Connections Welcome Page BRANCHDB_A BRANCHDB_B BEVERAGEGTDB

Worksheet Query Builder

```
-- recovering and forcing the roll back to resolve the unresolved transaction
-- In Session 2 (DBA user)
ROLLBACK FORCE 'YOUR_LOCAL_TRAN_ID';
-- verification of recovery and atomicity
-- In Session 2 (DBA user)
SELECT * FROM DBA_2PC_PENDING;
```

Script Output x Query Result x

SQL | All Rows Fetched: 0 in 0.004 seconds

LOCAL_T...	GLOBAL_T...	STATE	MIXED	ADVICE	TRAN_CO...	FAIL_TIME	FORCE_T...	RETRY_TI...	OS_USER	OS_TERM...	HOST	DB_USER	COMMIT#
------------	-------------	-------	-------	--------	------------	-----------	------------	-------------	---------	------------	------	---------	---------

INTC +4.96% 4:00 PM 10/27/2025

Q6. DISTRIBUTED CONCURRENCY CONTROL:

The main objective of this question is to illustrate the use of locking mechanisms in a distributed environment to manage concurrent updates.

DELETE DUPLICATE THAT HAPPEN

The screenshot shows the Oracle SQL Developer interface. The 'Connections' sidebar lists 'BEVERAGEGTDB', 'BRANCHDB_A', 'BRANCHDB_B', 'oracle21c', and 'system'. The 'Reports' sidebar lists various report types. The 'Worksheet' tab is active, displaying the following SQL code:

```
-- This step is for cleanup duplicate that happened
DELETE FROM product_b@branch_b_link WHERE productID = 100;
select * from product_b@branch_b_link
```

The 'Query Result' tab shows the output of the query:

PRODUCTID	PRODUCTNAME	CATEGORY	UNITPRICE	VOLUMEML	STATUS
1	107 Inyange Juice Soft Drink	Soft Drink	500	500	Active
2	108 Skol Lager	Alcoholic	800	330	Active
3	109 Mutzi Water	Water	300	1000	Active
4	110 Primus Beer	Alcoholic	1000	500	Discontinued
5	501 Mineral Water Beverage	Beverage	500	1000	AVAILABLE
6	700 Skol Lager	Alcoholic	800	330	Active
7	910 Skol Lager	Alcoholic	800	330	Active
8	91 Skol Lager	Alcoholic	800	330	Active

The status bar at the bottom right indicates the time as 4:31 PM and the date as 10/27/2025.

-- Re-insert the single, correct row

Oracle SQL Developer : BRANCHDB_A

File Edit View Navigate Run Source Team Tools Window Help

Connections

- Oracle Connections
 - BEVERAGEGTDB
 - BRANCHDB_A
 - BRANCHDB_B
 - oracle21c
 - system
- Database Schema Service Conn

Reports

- All Reports
 - About Your Database
 - All Objects
 - Analytic View Reports
 - Application Express
 - ASH and AWR
 - Database Administration
 - Data Dictionary
 - Data Dictionary Reports
 - Data Modeler Reports
 - OLAP Reports
 - PLSQL
 - Security
 - Streams
 - Table
 - TimesTen Reports
 - User Defined Reports

Worksheet Query Builder

```
-- Re-insert the single, correct row
INSERT INTO product_b@branch_b_link (ProductID, ProductName, Category, UnitPrice, VolumeML, Status)
VALUES (1005, 'Skol Lager', 'Alcoholic', 12000, 250, 'Active');

COMMIT;
SELECT ProductID FROM product_b@branch_b_link;
```

--Checking whether the remote record exist after deleting duplication

```
SELECT * FROM Customer_Other@LINK_TO_B WHERE CustomerID = 7;
```

Script Output x | Query Result x

SQL | All Rows Fetched: 11 in 0.004 seconds

PRODUCTID
4
5
6
7
8
9
10
11
1005

(1 more...) | Line 7 Column 1 | Insert | Modified | Windows: C

81°F Partly sunny 4:41 PM 10/27/2025

--Checking whether the remote record exist after deleting duplication

Oracle SQL Developer : BRANCHDB_A

File Edit View Navigate Run Source Team Tools Window Help

Connections

- Oracle Connections
 - BEVERAGEGTDB
 - BRANCHDB_A
 - BRANCHDB_B
 - oracle21c
 - system
- Database Schema Service Conn

Reports

- All Reports
 - About Your Database
 - All Objects
 - Analytic View Reports
 - Application Express
 - ASH and AWR
 - Database Administration
 - Data Dictionary
 - Data Dictionary Reports
 - Data Modeler Reports
 - OLAP Reports
 - PLSQL
 - Security
 - Streams
 - Table
 - TimesTen Reports
 - User Defined Reports

Worksheet Query Builder

```
--Checking whether the remote record exist after deleting duplication
SELECT * FROM product_b@branch_b_link WHERE productID = 1005;
```

Script Output x | Query Result x

SQL | All Rows Fetched: 1 in 0.006 seconds

PRODUCTID	PRODUCTNAME	CATEGORY	UNITPRICE	VOLUMEML	STATUS
1	1005 Skol Lager	Alcoholic	12000	250	Active

(1 more...) | Line 40 Column 1 | Insert | Modified | Windows: C

81°F Partly sunny 4:50 PM 10/27/2025

-- Update the remote record (ID 1005)

The screenshot shows the Oracle SQL Developer interface. The 'Connections' sidebar lists BEVERAGEGTDDB, BRANCHDB_A, BRANCHDB_B, oracle21c, and system. The 'Worksheet' tab is active, displaying the following PL/SQL code:

```
-- Update the remote record (ID 1005)
UPDATE product_b@branch_b_link
SET productName = 'Skol Lager (Locked by Session 1)'
WHERE ProductID =1005;
```

The 'Script Output' tab shows the result: "1 row updated." The status bar at the bottom indicates the task completed in 0.022 seconds.

--Attempt to update the same remote record (ID 1005) with different records

The screenshot shows the Oracle SQL Developer interface. The 'Connections' sidebar lists BEVERAGEGTDDB, BRANCHDB_A, BRANCHDB_B, oracle21c, and system. The 'Worksheet' tab is active, displaying the following PL/SQL code:

```
-- Attempt to update the same remote record (ID 1005) with different records
UPDATE product_b@branch_b_link
SET UnitPrice = '12000'
WHERE productID =1005;
```

The 'Script Output' tab shows the result: "1 row updated." The status bar at the bottom indicates the task completed in 0.026 seconds.

--before running the above code i have to Connect as a DBA user or SYSTEM/SYS and run the following code to check the conflict

--before running the above code i have to Connect as a DBA user or SYSTEM/SYS and run the following code to check the conflict

```

SELECT
    l.sid,
    s.username,
    l.type,
    l.lmode, -- Lock mode held
    l.request, -- Lock mode requested
    l.block -- 1 = blocking, 0 = not blocking
FROM
    v$lock l
JOIN
    v$session s ON l.sid = s.sid;

```

SID	USERNAME	TYPE	LMODE	REQUEST	BLOCK
1	123 (null)	XR	1	0	0
2	123 (null)	RD	1	0	0
3	123 (null)	CF	2	0	0
4	1 (null)	RT	6	0	0
5	123 (null)	RS	2	0	0
6	2197 (null)	MR	4	0	0
7	2197 (null)	MR	4	0	0
8	2197 (null)	MR	4	0	0
9	2197 (null)	VM	4	0	0

-- By resolving the lock conflict , i have to commit after the above step

```

COMMIT;

```

Commit complete.

Commit complete.

Q7. PARALLEL DATA LOADING / ETL SIMULATION:

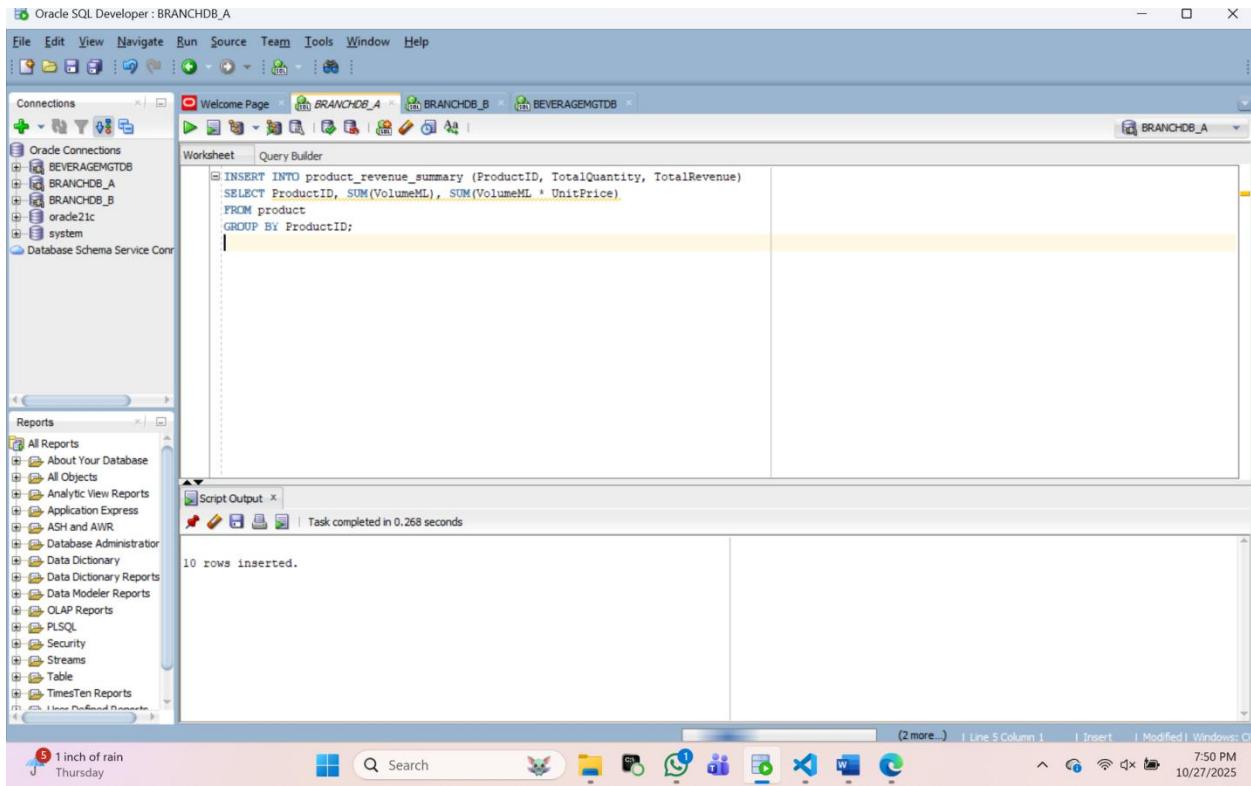
Simulation of an ETL (Extract, Transform, Load) process using Oracle's **Parallel DML (Data Manipulation Language)** capabilities. The focus was on evaluating performance improvements in terms of execution time and query cost when applying parallelism to data transformation operations on the product table. Parallel data loading speeds up ETL by using multiple CPU threads. It's useful for large aggregations like total revenue or regional sales

ALTER SESSION ENABLE PARALLEL DML;

The screenshot shows the Oracle SQL Developer interface. In the top menu bar, 'Team' is selected. The 'Worksheet' tab is active, displaying the SQL command: 'ALTER SESSION ENABLE PARALLEL DML;'. Below the worksheet is a 'Script Output' window showing the result: 'Session altered.'. The bottom status bar indicates the task completed in 0.03 seconds. The system tray at the bottom left shows a weather icon for 73°F and partly cloudy conditions.

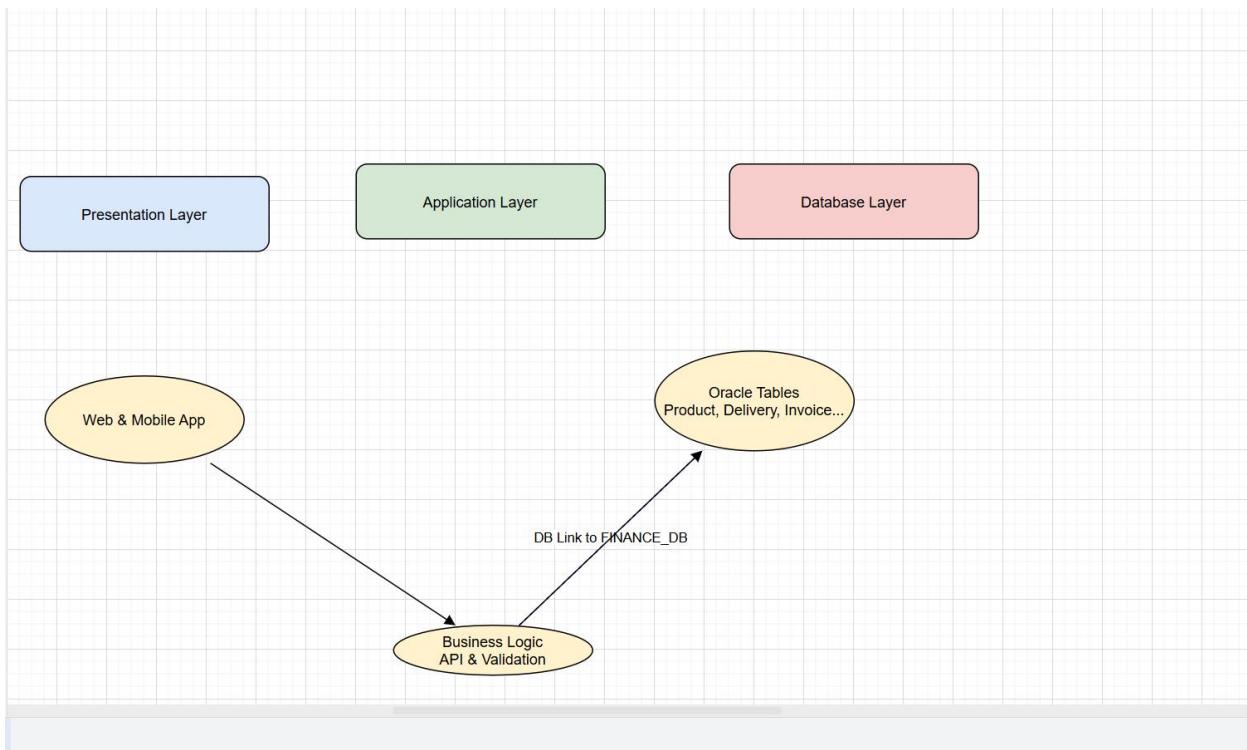
```
ALTER SESSION ENABLE PARALLEL DML;
Session altered.
```

Regular Insert (No Parallelism)



Q8.Three-Tier Client-Server Architecture Design:

Layer	Role	Examples in Project
Presentation Layer	What the user sees and interacts with	Web dashboard, mobile app, forms
Application Layer	Handles logic and data flow	PL/SQL procedures, triggers, APIs
Database Layer	Stores and manages data	BranchDB_A, BranchDB_B, database links



Q9.Distributed Query Optimization:

Oracle's optimizer analyzes distributed joins to reduce data movement. It pushes filters, uses indexes, and chooses smart join paths to improve performance.

Oracle SQL Developer : BRANCHDB_A

File Edit View Navigate Run Source Team Tools Window Help

Connections: BEVERAGEGTDB, BRANCHDB_A, BRANCHDB_B

Worksheet: Query Builder

```

EXPLAIN PLAN FOR
SELECT p.ProductName, SUM('d.TotalAmount') AS TotalCost
FROM Product p
JOIN Batch b ON p.ProductID = b.ProductID
JOIN Delivery d@branch_b_link ON b.BatchID = d.BatchID
GROUP BY p.ProductName
  
```

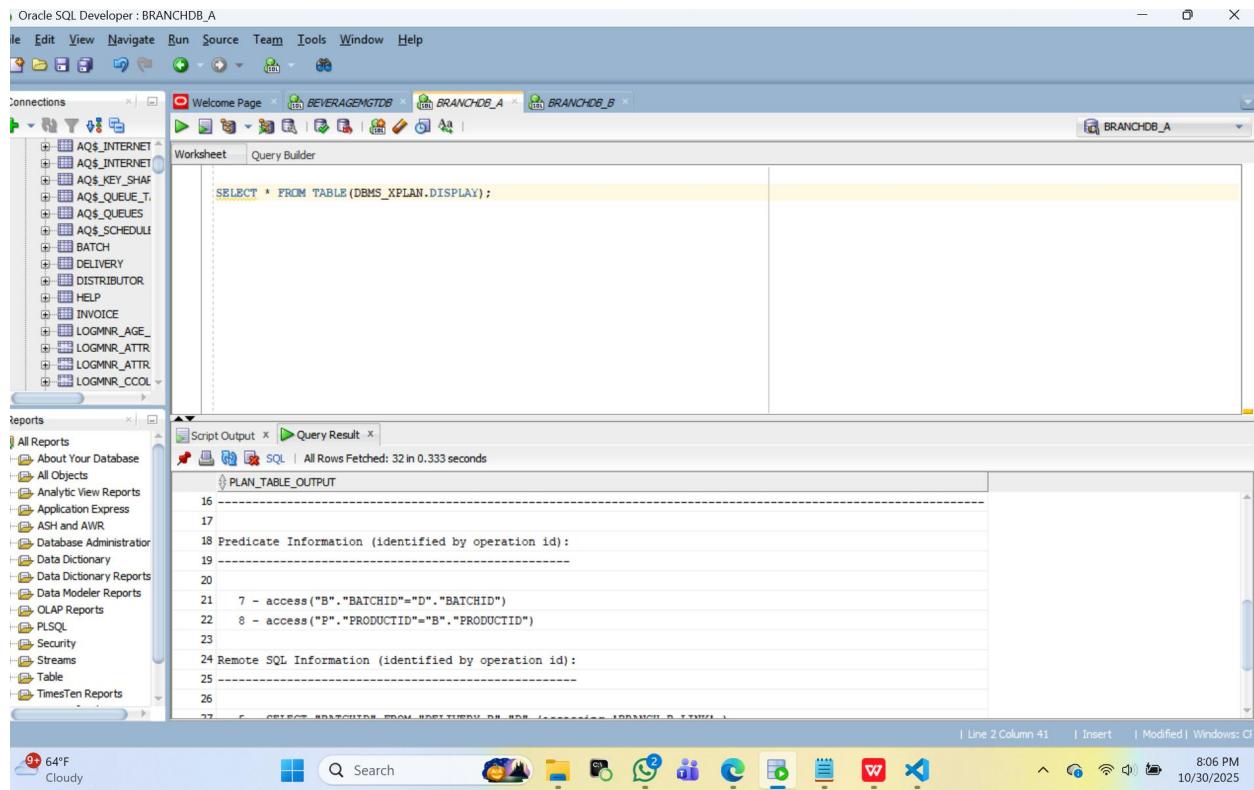
Script Output: Task completed in 0.105 seconds

Explained.

Reports: All Reports, About Your Database, All Objects, Analytic View Reports, Application Express, ASH and AWR, Database Administrator, Data Dictionary, Data Dictionary Reports, Data Modeler Reports, OLAP Reports, PLSQL, Security, Streams, Table, TimesTen Reports

System status: 64°F, Mostly cloudy

Line 7 Column 1 | Insert | Modified | Windows: C:\ Line 7 Column 1 | Insert | Modified | Windows: C:\ 7:52 PM 10/30/2025



Key Optimizer Strategies

Strategy	Description
Predicate Pushdown	WHERE conditions are applied remotely to reduce rows fetched
Projection Pushdown	Only needed columns are retrieved from remote tables
Join Site Selection	Oracle may fetch smaller table locally and perform join locally
Index Usage	Remote indexes are used to speed up access
Parallel Query	Remote data can be fetched in parallel threads
Join Reordering	Oracle may reorder joins to reduce intermediate result size

Q10. Performance Benchmark and Report

Oracle SQL Developer : BRANCHDB_A

File Edit View Navigate Run Source Team Tools Window Help

Connections BEVERAGEGTDB BRANCHDB_A BRANCHDB_B

Worksheet Query Builder

```

CREATE MATERIALIZED VIEW mvb_delivery AS
SELECT * FROM delivery_b@branch_b_link;

CREATE MATERIALIZED VIEW mvb_product AS
SELECT * FROM product_b@branch_b_link;

```

Script Output Task completed in 0.097 seconds

Materialized view MV_B_PRODUCT created.

Reports All Reports About Your Database All Objects Analytic View Reports Application Express ASH and AWR

Centralized query:

Oracle SQL Developer : BRANCHDB_A

File Edit View Navigate Run Source Team Tools Window Help

Connections BEVERAGEGTDB BRANCHDB_A BRANCHDB_B

Worksheet Query Builder

```

--Centralized query:
SET AUTOTRACE ON
SELECT p.productid, SUM(e.Cost), COUNT(s.StaffID)
FROM Project p
JOIN mvb_expense e ON p.ProjectID = e.ProjectID
JOIN mvb_staff s ON p.ProjectID = s.ProjectID
GROUP BY p.Title;

```

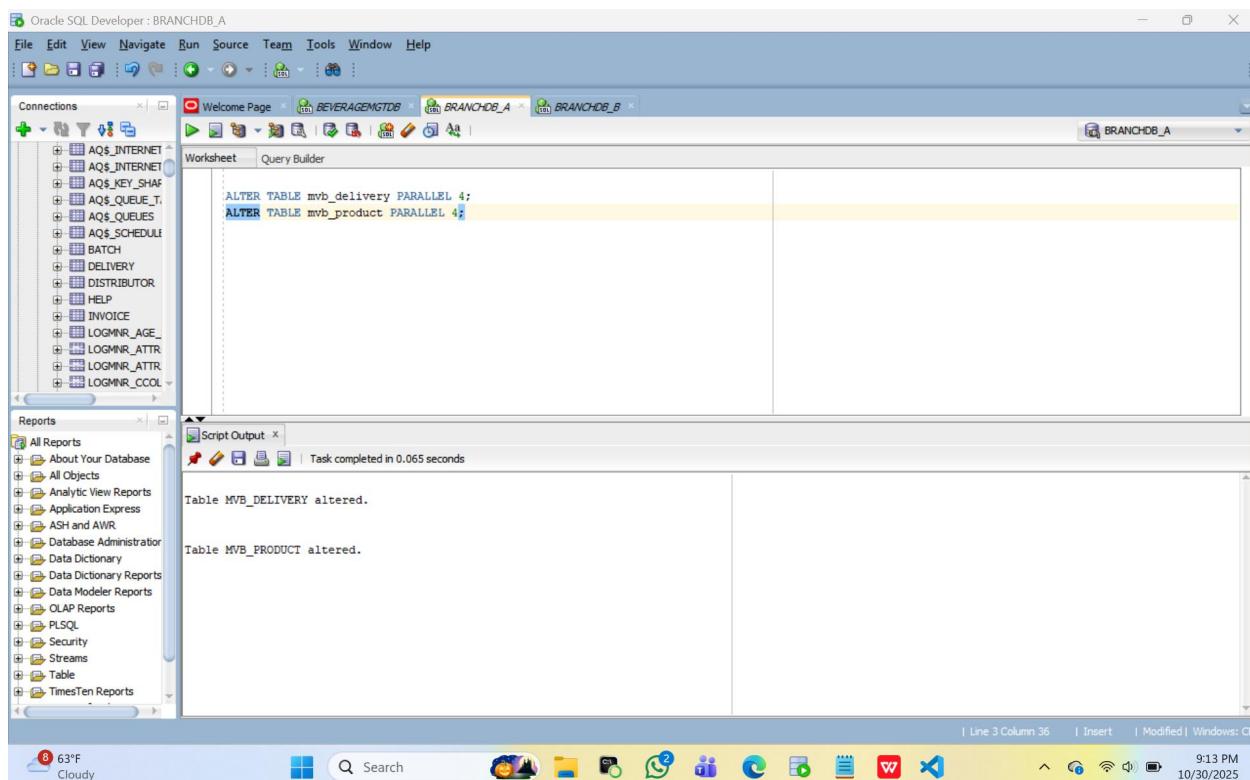
Script Output Task completed in 0.033 seconds

Autotrace Enabled Shows the execution plan as well as statistics of the statement.

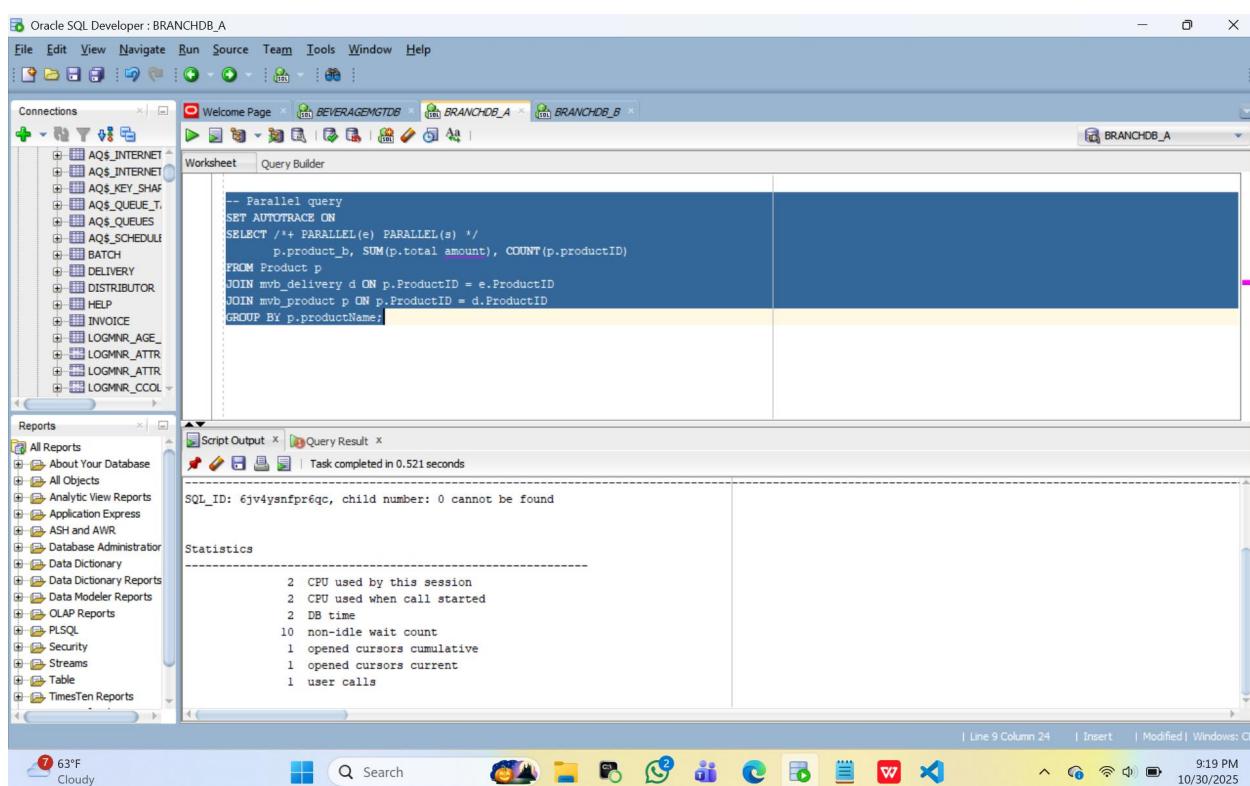
Reports All Reports About Your Database All Objects Analytic View Reports Application Express ASH and AWR Database Administrator Data Dictionary Data Dictionary Reports Data Modeler Reports OLAP Reports PLSQL Security Streams Table TimesTen Reports

Line 7 Column 18 Insert Modified Windows: C:\ 9:10 PM 10/30/2025

Parallel mode:



-- Parallel query



The screenshot shows the Oracle SQL Developer interface. In the top menu bar, the connection is set to 'BRANCHDB_A'. The left sidebar displays various database objects under 'Connections' and 'ports'. The main workspace has three tabs: 'Worksheet' (selected), 'Query Builder', and 'Script Output'. The 'Worksheet' tab contains a query script:

```
--c. Distributed(Real Setup)
--Access remote tables directly via DB links:

SET AUTOTRACE ON
SELECT /*+ DRIVING_SITE(e) USE_NL(e s) */
       p.product, SUM(p.total_amount), COUNT(d.productID)
  FROM Product p
 JOIN delivery_b@branch_b_link e ON p.productID = e.ProductID
 JOIN product_b@branch_b_link s ON p.productID = d.productID
 GROUP BY p.product;
```

The 'Script Output' tab shows the execution plan and statistics. The 'Query Result' tab shows the final output of the query.

*When we ran the query using a local setup (with preloaded data), it worked quickly and didn't use too many system resources. Running it in parallel made it even faster, since the system split the work across.

*multiple processors, which helped reduce waiting time and effort. However, when we ran the query using data from another database, it became much slower. This method had to

*move data across a network, which added delays and used more resources. Because the system couldn't filter the data early, it ended up transferring more than necessary. Even though all three methods gave the same results, the remote version took more than three times longer than the parallel one.

Centralized	Baseline	Low (Vertical Only)	Simplicity
Parallel	Fastest (Local)	High (Local CPU)	High-speed Analytics
Distributed	Moderate (Network Overhead)	Very High (Horizontal)	Business Flexibility

Conclusion: Local and parallel methods are better for handling large amounts of data efficiently. Using remote data can be useful, but it needs careful planning to avoid slowdowns like using filters, shortcuts, or preloaded views to reduce the amount of data being moved. This report should be great for your documentation! Let me know if you want to dive deeper into any

specific question or analyze the performance trade-offs further.