



## DSA 2040A US 2025 LAB 2

---

### Lab Manual: EXTENDED LAB MANUAL: OLTP + OLAP Integration

#### Objective

Simulate a real-world retail system by:

- Designing an OLTP schema (for daily operations)
- Building an OLAP star schema (for analytics)
- Performing ETL from OLTP → OLAP
- Running analytical queries to gain insights

#### STEP 1 – OLTP: Design the Transactional Database

##### Purpose:

This is where raw, real-time business data like **sales transactions** and **customer information** is recorded.

##### Tables to Create:

##### 1. customers table – Stores customer info

```
CREATE TABLE customers (  
    customer_id SERIAL PRIMARY KEY,  
    name VARCHAR(100),  
    email VARCHAR(100) UNIQUE,  
    registered_date DATE  
);
```

**Explanation:** Each customer has a unique ID. We track their name, email, and when they registered.

##### 2. products table – Holds product catalog

```
CREATE TABLE products (  
    product_id SERIAL PRIMARY KEY,  
    name VARCHAR(100),  
    category VARCHAR(50),  
    price DECIMAL(10,2)
```

```
);
```

**Explanation:** This stores the available products with category and price.

### 3. stores table – Contains store details

```
CREATE TABLE stores (  
    store_id SERIAL PRIMARY KEY,  
    name VARCHAR(100),  
    region VARCHAR(50)  
);
```

**Explanation:** Reflects where the products are sold — useful for regional sales analysis.

### 4. transactions table – Records sales activity

```
CREATE TABLE transactions (  
    transaction_id SERIAL PRIMARY KEY,  
    customer_id INT REFERENCES customers(customer_id),  
    product_id INT REFERENCES products(product_id),  
    store_id INT REFERENCES stores(store_id),  
    quantity INT NOT NULL,  
    transaction_date DATE NOT NULL  
);
```

**Explanation:** This is the **main OLTP fact table**, capturing who bought what, how many, where, and when.

## STEP 2 – Insert Sample Data

**Use these SQL commands to simulate data entry:**

```
-- Customers  
INSERT INTO customers (name, email, registered_date)  
VALUES  
('Alice Smith', 'alice@example.com', '2025-01-10'),  
('Bob Brown', 'bob@example.com', '2025-02-01');  
  
-- Products  
INSERT INTO products (name, category, price)  
VALUES  
('T-Shirt', 'Apparel', 19.99),  
('Laptop', 'Electronics', 999.00);  
  
-- Stores  
INSERT INTO stores (name, region)  
VALUES  
('Downtown Store', 'East'),
```

```
('Mall Outlet', 'West');
```

```
-- Transactions
```

```
INSERT INTO transactions (customer_id, product_id, store_id, quantity,  
transaction_date)
```

```
VALUES
```

```
(1, 1, 1, 2, '2025-05-01'),
```

```
(2, 2, 2, 1, '2025-05-03');
```

### **STEP 3 – OLAP: Build the Data Warehouse (Star Schema)**

#### **Tables from the original lab:**

- dim\_date
- dim\_product
- dim\_store
- fact\_sales

These are **denormalized** and optimized for analytical queries.

#### **Use the existing schema in the lab document:**

```
-- Example: fact_sales
```

```
CREATE TABLE fact_sales (
```

```
    product_id INT,
```

```
    store_id INT,
```

```
    date_id INT,
```

```
    quantity_sold INT,
```

```
    revenue DECIMAL(10,2)
```

```
);
```

### **STEP 4 – ETL Process: From OLTP to OLAP**

#### **Purpose:**

Convert normalized transaction data into an analytical format.

#### **SQL ETL Script:**

```
-- ETL to populate fact_sales
```

```
INSERT INTO fact_sales (product_id, store_id, date_id, quantity_sold, revenue)
```

```
SELECT
```

```
    t.product_id,
```

```
    t.store_id,
```

```
    d.date_id,
```

```
    t.quantity,
```

```
    t.quantity * p.price AS revenue
```

```
FROM
```

```

    transactions t
JOIN
    products p ON t.product_id = p.product_id
JOIN
    dim_date d ON t.transaction_date = d.full_date;

```

### What happens here?

- Joins OLTP data with product price and date dimension
- Calculates revenue
- Loads it into the OLAP star schema

## STEP 5 – Run OLAP Analytical Queries

### Example 1: Monthly Sales Trends

```

SELECT d.month, SUM(fs.revenue) AS total_revenue
FROM fact_sales fs
JOIN dim_date d ON fs.date_id = d.date_id
GROUP BY d.month
ORDER BY d.month;

```

### Example 2: Top Selling Products

```

SELECT dp.name, SUM(fs.quantity_sold) AS total_units
FROM fact_sales fs
JOIN dim_product dp ON fs.product_id = dp.product_id
GROUP BY dp.name
ORDER BY total_units DESC
LIMIT 5;

```

## STEP 6 – Final Discussion

### Reflection Questions:

1. Why is the OLTP system normalized and the OLAP system denormalized?
2. What challenges would you face if you ran analytical queries directly on the OLTP system?
3. How can automation (e.g., scheduled ETL jobs) help in a real-world data pipeline?

### Project Organization (GitHub Structure)

```

/oltp_schema/      -- Contains OLTP creation scripts
/olap_schema/      -- Contains star schema SQL
/etl_scripts/      -- SQL to perform ETL
/sample_data/      -- Optional CSVs for seeding

```

```
/queries/      -- Analytical queries
README.md      -- Document objectives, steps, and team info
```