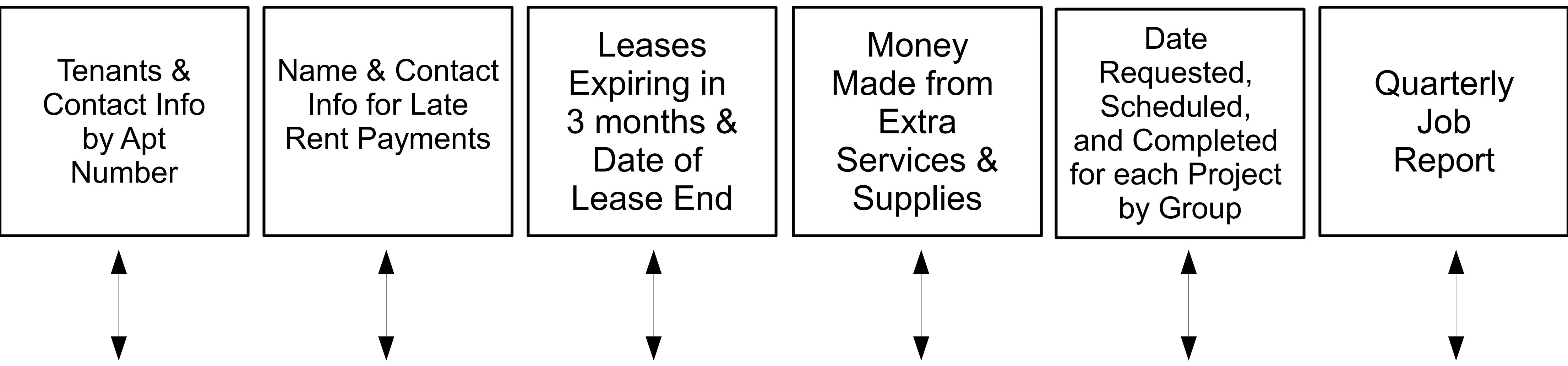
Apartment Complex 3 Schema

Andrew Ebert


Tenants & Contact Info by Apt Number
Name & Contact Info for Late Rent Payments
Leases Expiring in 3 months & Date of Lease End
Money Made from Extra Services & Supplies
Date Requested, Scheduled, and Completed for each Project by Group
Quarterly Job Report

APARTMENT

Apt_no	No_bedrooms	Monthly_rent

TENANT

Tenant_id	Contact_info	Name	Lease_id

MAINTENANCE PROJECT

Job_id	Date_requested	Date_scheduled	Date_completed	Maintenance_type	Apt_no

LEASE

Lease_id	Term	Apt_no	Release

AMENITIES

Amenity_id	Type	Daily_fee	Price	Monthly_rent	Count

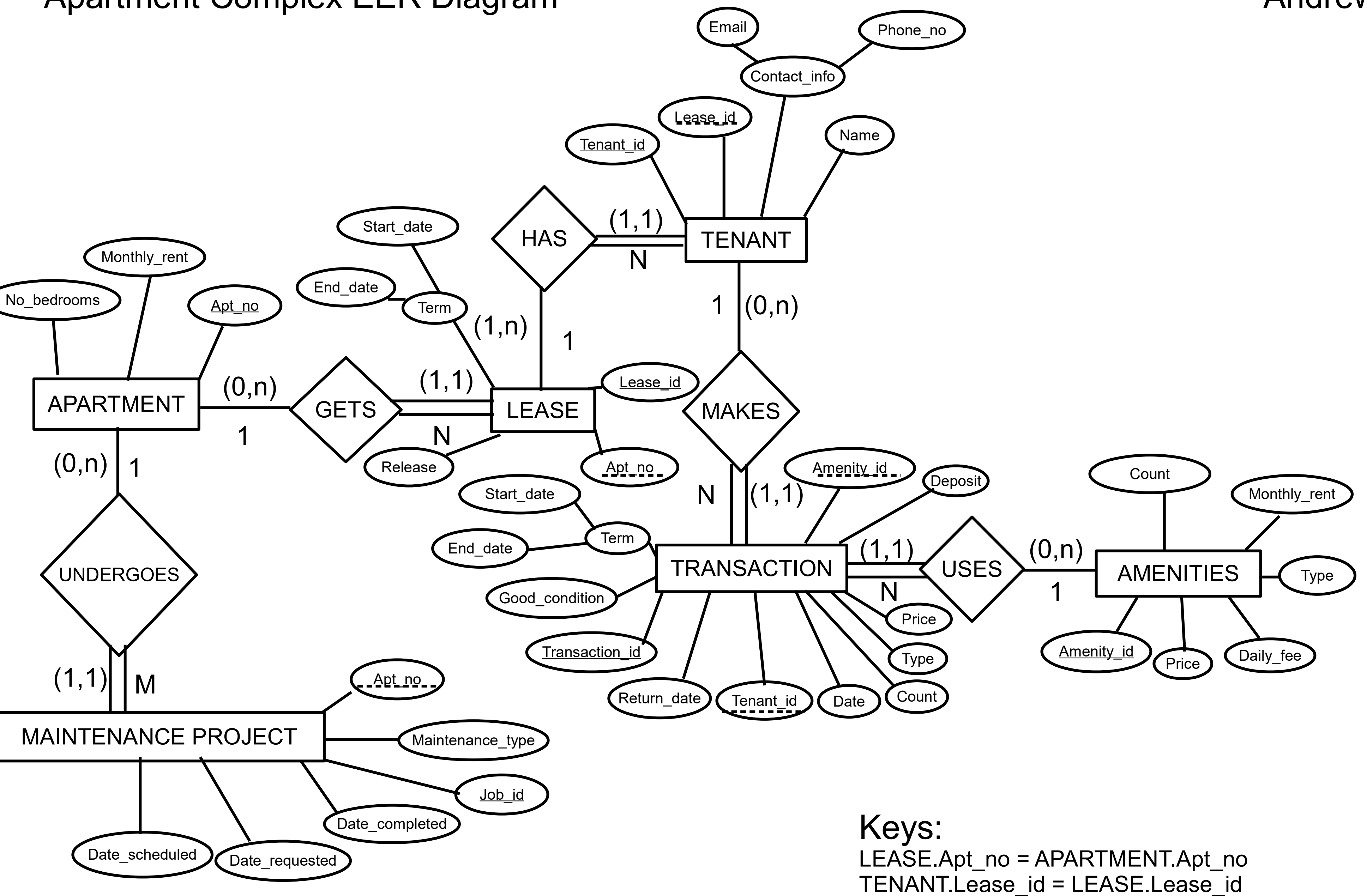
TRANSACTION

Transaction_id	Tenant_id	Type	Date	Amenity_id	Deposit	Count	Return_date	Good_condition	Price	Term

Server Name:	csci3287.cse.ucdenver.edu
PHPMyAdmin:	https://csci3287.cse.ucdenver.edu/phpmyadmin
Webpage:	https://csci3287.cse.ucdenver.edu/~eberta/db
Username:	eberta
Pwd:	DLcJJNyoyjqF
DBName =	eberta_DB

Apartment Complex EER Diagram

Andrew Ebert



Keys:

LEASE.Apt_no = APARTMENT.Apt_no
TENANT.Lease_id = LEASE.Lease_id
TRANSACTION.Tenant_id = TENANT.Tenant_id
TRANSACTION.Amenity_id = AMENITIES.Amenity_id
MAINTENANCE_PROJECT.Apt_no = APARTMENT.Apt_no

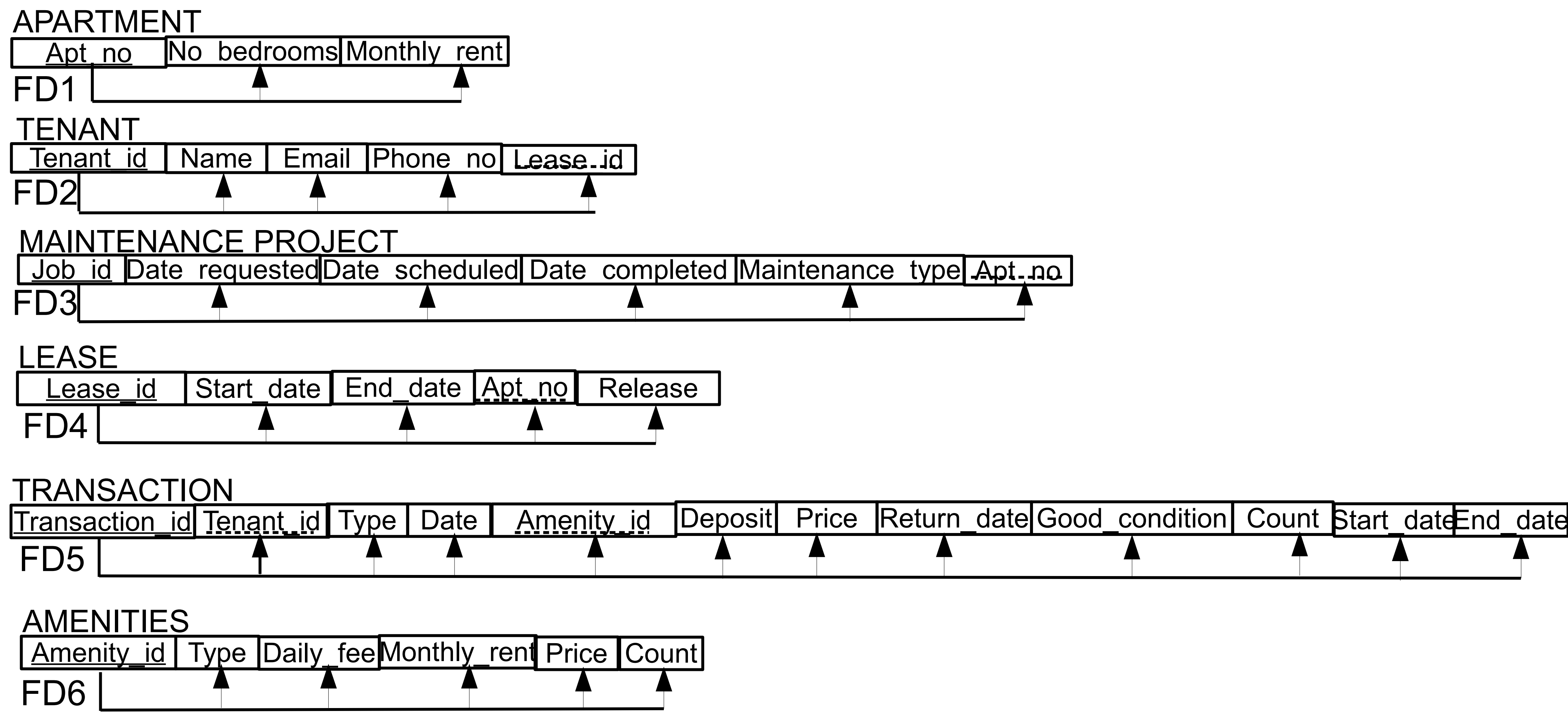
Assumptions:

A TENANT must have a LEASE to be entered into the Database.
A TENANT can only have one LEASE but a LEASE can have many TENANTS.
A LEASE must have at least one TENANT.
An APARTMENT can get LEASED many times but a LEASE can only have one APARTMENT.
A MAINTENANCE PROJECT must have one APARTMENT that is receiving maintenance in order to be entered into the database.
An APARTMENT can have many MAINTENANCE PROJECTS but a MAINTENANCE PROJECT can only have one APARTMENT.
A TRANSACTION must have a TENANT and AMENITY in order to be entered into the database.
A TENANT can make many TRANSACTIONS, but a TRANSACTION can only be made by one TENANT.
An AMENITY can be apart of many TRANSACTIONS, but a TRANSACTION can only have one AMENITY.
An APARTMENT doesn't need to be LEASED ever.
A TENANT doesn't need to make a TRANSACTION .
An AMENITY doesn't need to be apart of a TRANSACTION.

TENANT.Email and TENANT.Phone_no aren't primary keys, but they are unique to each TENANT.

Apartment Complex 3NF Diagram & Functional Dependencies

Andrew Ebert



Apartment Complex Relational Algebra Andrew Ebert

Q1 ⟵ πLEASE.Apt_no, TENANT.Name, TENANT.Email, TENANT.Phone_no **Order by apt_no
(LEASE⋈LEASE.Lease_id = TENANT.Lease_idTENANT)

LATEST TRANSACTION “Tenant_id”, “Latest_rent_date” ⟵ Tenant_id ℑ MAXIMUM(TRANSACTION.Date)
(TRANSACTION.Type = “Rent”
(TRANSACTION⋈TRANSACTION.Tenant_id = TENANT.Tenant_id TENANT)

Q2⟵ TENANT.Tenant_id, TENANT.Name, TENANT. Email, TENANT.Phone_no, LEASE.Start_date, LEASE.End_date, LATEST TRANSACTION.Latest_rent_date,
((MONTH(CURRENT_DATE)) - MONTH(LATEST TRANSACTION.Latest_rent_date)) * APARTMENT.Monthly_rent AS Rent_owed
(σLATEST TRANSACTION.Latest_rent_date < 12/1/2023
(APARTMENT⋈APARTMENT.Apt_no = LEASE.Apt_noLEASE⋈LEASE.Lease_id = TENANT.Lease_id TENANT⋈TENANT.Tenant_id = LATEST TRANSACTION.Tenant_id LATEST TRANSACTION))

Q3 ⟵ πLEASE.Lease_id, LEASE.End_date, LEASE.Release, TENANT.Name, TENANT.Email, TENANT.Phone_no,
(σLEASE.End_date <= (CURRENT DATE + INTERVAL 3 MONTH)
(LEASE⋈LEASE.Lease_id = TENANT.Lease_id TENANT))

GROUPED“Type”, “Month”, “Money_made” ⟵ Type, MONTH(Date) ℑ SUM(Price)
(TRANSACTION)

Q4 ⟵ πType, Month, Money_made **Order by type & month
(σ(Type = “Loan” OR Type = “Purchase”) AND YEAR(Date) = YEAR(CURDATE())
(GROUPED))

Q5 ⟵ πMaintenance_type, Job_id,Date_requested, Date_scheduled, Date_completed **Order by type
(MAINTENANCE PROJECT)

TOTAL PROJECTS“Maintenance_type”, “Total” ⟵ Maintenance_type ℑ COUNT(MAINTENANCE PROJECT.Maintenance_type)
(σMAINTENANCE PROJECT.Date_requested >= (CURRENT DATE - INTERVAL 3 MONTH)
(MAINTENANCE PROJECT))

COMPLETED PROJECTS“Maintenance_type”, “Completed” ⟵ Maintenance_type ℑ COUNT(MAINTENANCE PROJECT.Maintenance_type)
(σMAINTENANCE PROJECT.Date_requested >= (CURRENT DATE - INTERVAL 3 MONTH) AND MAINTENANCE PROJECT.Date_completed IS NOT NULL
(MAINTENANCE PROJECT))

TIME TO COMPLETE“Maintenance_type”, “Avg_hrs_to_complete” ⟵ Maintenance_type ℑ AVERAGE((TIMESTAMPDIFF(HOUR, MAINTENANCE PROJECT.Date_requested, MAINTENANCE PROJECT.Date_completed)))
(σMAINTENANCE PROJECT.Date_requested >= (CURRENT DATE - INTERVAL 3 MONTH) AND MAINTENANCE PROJECT.Date_completed IS NOT NULL
(MAINTENANCE PROJECT))

Q6⟵ πMaintenance_type, Total, Completed, (Completed / Total) * 100 AS Percentage, Avg_hrs_to_complete
(TOTAL PROJECTS⋈TOTAL PROJECTS.Maintenance_type = COMPLETED PROJECTS.Maintenance_type COMPLETED PROJECTS⋈COMPLETED PROJECTS.Maintenance_type = HOURS TO COMPLETE.Maintenance_typeHOURS TO COMPLETE)

```
<!DOCTYPE html>
<html>
<head>
    <style>
        body {
            background-color: #CFB87C;
        }
        h1 {
            text-align: center;
            font-family: "Arial";
            font-weight: bold;
            text-transform: uppercase;
        }
        p {
            text-align: center;
            font-family: "Arial";
        }
        table {
            border-collapse: collapse;
        }
        th, td {
            border: 1px solid black;
            padding: 8px;
            text-align: left;
        }
        th {
            background-color: #f2f2f2;
        }
    </style>
    <title>Welcome to your Database Page</title>
</head>
<body>
    <h1>Database Page for Apartment Complex CSCI 3287</h1>

    <?php
    $servername = "csci3287.cse.ucdenver.edu";
    $username = "eberta";
    $password = "DLcJJNyoyjqF";
    $dbname = "eberta_DB";

    try {
        $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
        $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

        //first query
        $sql = "SELECT LEASE.Apt_no, TENANT.Name, TENANT.Email, TENANT.Phone_no
                FROM LEASE
                JOIN TENANT ON LEASE.Lease_id = TENANT.Lease_id
                ORDER BY LEASE.Apt_no";
        $result = $conn->query($sql);

        if ($result->rowCount() > 0) {
            echo "<h2>Tenants & Contact Info by Apartment Number</h2>";
            echo "<table>";
            echo "<tr><th>Apt_no</th><th>Name</th><th>Email</th><th>Phone_no</th></tr>";
            while ($row = $result->fetch(PDO::FETCH_ASSOC)) {
                echo "<tr><td>".$row['Apt_no']."</td><td>".$row['Name']."</td><td>".$row['Email']."</td><td>".
$row['Phone_no']."</td></tr>";
            }
            echo "</table>";
        } else {
            echo "0 results";
        }
        //second query
        $sql2 = "SELECT TENANT.Tenant_id, TENANT.Name, TENANT.Email, TENANT.Phone_no, LEASE.Start_date, LEASE.End_date,
((MONTH(CURRENT_DATE)) - MONTH(MAX(TRANSACTION.Date))) * APARTMENT.Monthly_rent AS Rent_owed, MAX(TRANSACTION.Date) AS
Last_payment
                FROM TENANT
                JOIN LEASE ON TENANT.Lease_id = LEASE.Lease_id
                JOIN APARTMENT ON LEASE.Apt_no = APARTMENT.Apt_no
                LEFT JOIN TRANSACTION ON TENANT.Tenant_id = TRANSACTION.Tenant_id
                WHERE TRANSACTION.Type = 'Rent' AND TRANSACTION.Date < LEASE.End_date
                GROUP BY TENANT.Tenant_id, TENANT.Name, TENANT.Email, TENANT.Phone_no, LEASE.Start_date, LEASE.End_date,
APARTMENT.Monthly_rent
                HAVING MAX(TRANSACTION.Date) < '2023-12-01'";
        $result2 = $conn->query($sql2);

        if ($result2->rowCount() > 0) {
            echo "<h2>Name & Contact Info for Late Rent Payments with Last Payment</h2>";
            echo "<table>";
            echo
"<tr><th>Tenant_id</th><th>Name</th><th>Email</th><th>Phone_no</th><th>Start_date</th><th>End_date</th><th>Rent_owed</th><t
h>Last_payment</th></tr>";
            while ($row = $result2->fetch(PDO::FETCH_ASSOC)) {
                echo "<tr><td>".$row['Tenant_id']."</td><td>".$row['Name']."</td><td>".$row['Email']."</td><td>".
$row['Phone_no']."</td><td>".$row['Start_date']."</td><td>".$row['End_date']."</td><td>".$row['Rent_owed']."</td><td>".
$row['Last_payment']."</td></tr>";
            }
            echo "</table>";
        } else {
            echo "0 results";
        }

        //third query
        $sql3 = "SELECT LEASE.Lease_id, LEASE.End_date, LEASE.Release, TENANT.Name, TENANT.Email, TENANT.Phone_no
                FROM LEASE, TENANT
                WHERE LEASE.Lease_id = TENANT.Tenant_id AND LEASE.End_date <= (CURRENT_DATE + INTERVAL 3 MONTH)";
        $result3 = $conn->query($sql3);

        if ($result3->rowCount() > 0) {
            echo "<h2>Leases Expiring in 3 months & Date of Lease End</h2>";
            echo "<table>";
            echo "<tr><th>Lease_id</th><th>End_date</th><th>Release</th><th>Name</th><th>Email</th><th>Phone_no</th></tr>";
            while ($row = $result3->fetch(PDO::FETCH_ASSOC)) {
                echo "<tr><td>".$row['Lease_id']."</td><td>".$row['End_date']."</td><td>".$row['Release']."</td><td>".
$row['Name']."</td><td>".$row['Email']."</td><td>".$row['Phone_no']."</td></tr>";
            }
            echo "</table>";
        } else {
            echo "0 results";
        }

        //fourth query
        $sql4 = "SELECT Type, MONTH(Date) AS Transaction_month, SUM(Price) AS Money_made
                FROM TRANSACTION
                WHERE (Type = 'Loan' or Type = 'Purchase' AND YEAR(Date) = YEAR(CURDATE()))
                GROUP BY Type, Transaction_month
                ORDER BY Type, Transaction_month";
        $result4 = $conn->query($sql4);
        if ($result4->rowCount() > 0) {
            echo "<h2>Money Made from Extra Services & Supplies</h2>";
            echo "<table>";
            echo "<tr><th>Type</th><th>Transaction_month</th><th>Money_made</th></tr>";
            while ($row = $result4->fetch(PDO::FETCH_ASSOC)) {
                echo "<tr><td>".$row['Type']."</td><td>".$row['Transaction_month']."</td><td>".
$row['Money_made']."</td></tr>";
            }
            echo "</table>";
        } else {
            echo "0 results";
        }




        //fifth query
        $sql5 = "SELECT Maintenance_type,Job_id,Date_requested,Date_scheduled,Date_completed
                FROM `MAINTENANCE PROJECT`
                ORDER BY Maintenance_type";
        $result5 = $conn->query($sql5);

        if ($result5->rowCount() > 0) {
            echo "<h2>Date Requested, Scheduled, and Completed for each Project by Group</h2>";
            echo "<table>";
            echo
"<tr><th>Maintenance_type</th><th>Job_id</th><th>Date_requested</th><th>Date_scheduled</th><th>Date_completed</th></tr>";
            while ($row = $result5->fetch(PDO::FETCH_ASSOC)) {
                echo "<tr><td>".$row['Maintenance_type']."</td><td>".$row['Job_id']."</td><td>".
$row['Date_requested']."</td><td>".$row['Date_scheduled']."</td><td>".$row['Date_completed']."</td></tr>";
            }
            echo "</table>";
        } else {
            echo "0 results";
        }

        //sixth query
        $sql6 = "SELECT Maintenance_type, COUNT(*) AS Total_projects, SUM(CASE WHEN `MAINTENANCE PROJECT`.Date_completed IS
NOT NULL THEN 1 ELSE 0 END) AS Projects_completed, (SUM(CASE WHEN `MAINTENANCE PROJECT`.`Date_completed` IS NOT NULL THEN 1
ELSE 0 END) / COUNT(*) * 100) AS Percentage_completed, (SUM(TIMESTAMPDIFF(HOUR, `MAINTENANCE PROJECT`.`Date_requested`,
`MAINTENANCE PROJECT`.`Date_completed`)) / SUM(CASE WHEN `MAINTENANCE PROJECT`.`Date_completed` IS NOT NULL THEN 1 ELSE 0
END)) AS Average_hours_to_complete
                FROM `MAINTENANCE PROJECT`
                WHERE `MAINTENANCE PROJECT`.`Date_requested` >= (CURRENT_DATE - INTERVAL 3 MONTH)
                GROUP BY `MAINTENANCE PROJECT`.`Maintenance_type`";
        $result6 = $conn->query($sql6);

        if ($result6->rowCount() > 0) {
            echo "<h2>Quarterly Job Report</h2>";
            echo "<table>";
            echo
"<tr><th>Maintenance_type</th><th>Total_projects</th><th>Projects_completed</th><th>Percentage_completed</th><th>Average_ho
urs_to_complete</th></tr>";
            while ($row = $result6->fetch(PDO::FETCH_ASSOC)) {
                echo "<tr><td>".$row['Maintenance_type']."</td><td>".$row['Total_projects']."</td><td>".
$row['Projects_completed']."</td><td>".$row['Percentage_completed']."</td><td>".
$row['Average_hours_to_complete']."</td></tr>";
            }
            echo "</table>";
        } else {
            echo "0 results";
        }



    } catch(PDOException $e) {
        echo "Connection failed: " . $e->getMessage();
    }
    $conn = null;
    ?>
</body>
</html>
```