```
-- Drop existing foreign key constraint


-- Add a new foreign key constraint
ALTER TABLE employee
ADD CONSTRAINT fk_departmentID
FOREIGN KEY (departmentID)
REFERENCES departments(departmentID);


select distinct clientid from projects

ALTER TABLE departments
ADD CONSTRAINT departmentID PRIMARY KEY (departmentID);

SELECT constraint_name, constraint_type
FROM user_constraints
WHERE table_name = 'projects';


SELECT constraint_name, column_name
FROM user_cons_columns
WHERE table_name = 'projects'
  AND constraint_name IN (
    SELECT constraint_name
    FROM user_constraints
    WHERE constraint_type = 'F'
  );



select * from projects p
join clients c
on p.clientid = c.clientid

ALTER TABLE Projects
ADD CONSTRAINT pk_project PRIMARY KEY (ProjectID);


ALTER TABLE Events
ADD CONSTRAINT fk_event_project
FOREIGN KEY (ProjectID)
REFERENCES Projects (ProjectID);

ALTER TABLE Employee
ADD CONSTRAINT pk_employee PRIMARY KEY (EmployeeID);

ALTER TABLE Events
ADD CONSTRAINT pk_event PRIMARY KEY (EventID);

ALTER TABLE Media_Contacts
ADD CONSTRAINT pk_mediacontact PRIMARY KEY (ContactID);

ALTER TABLE Contents
```

```
ADD CONSTRAINT pk_content PRIMARY KEY (ContentID);


ALTER TABLE PROJECTEMPLOYEE
ADD CONSTRAINT pk_projectid PRIMARY KEY (PROJECTID);

ALTER TABLE Invoice_Items
ADD CONSTRAINT pk_invoiceitem PRIMARY KEY (ItemID);

ALTER TABLE Payment
ADD CONSTRAINT pk_payment PRIMARY KEY (PaymentID);


ALTER TABLE Contracts
ADD CONSTRAINT pk_contract PRIMARY KEY (ContractID);

ALTER TABLE Suppliers
ADD CONSTRAINT pk_supplier PRIMARY KEY (SupplierID);

ALTER TABLE Purchase_Orders
ADD CONSTRAINT pk_purchaseorder PRIMARY KEY (POID);

ALTER TABLE Expenses
ADD CONSTRAINT pk_expense PRIMARY KEY (ExpenseID);

ALTER TABLE Resources
ADD CONSTRAINT pk_resource PRIMARY KEY (ResourceID);

ALTER TABLE Campaigns
ADD CONSTRAINT pk_campaign PRIMARY KEY (CampaignID);

ALTER TABLE Strategies
ADD CONSTRAINT pk_strategy PRIMARY KEY (StrategyID);

ALTER TABLE Campaign_Media
ADD CONSTRAINT pk_campaignmedia PRIMARY KEY (MediaID);

ALTER TABLE Target_Audience
ADD CONSTRAINT pk_targetaudience PRIMARY KEY (AudienceID);

ALTER TABLE Performance_Metrics
ADD CONSTRAINT pk_performancemetrics PRIMARY KEY (MetricID);

ALTER TABLE Feedback
ADD CONSTRAINT pk_feedback PRIMARY KEY (FeedbackID);

ALTER TABLE Contents
ADD CONSTRAINT fk_content_project
FOREIGN KEY (ProjectID)
REFERENCES Projects (ProjectID);

ALTER TABLE Billing
ADD CONSTRAINT fk_billing_client
FOREIGN KEY (ClientID)
```

```
REFERENCES Clients (ClientID);

ALTER TABLE Billing
ADD CONSTRAINT fk_billing_project
FOREIGN KEY (ProjectID)
REFERENCES Projects (ProjectID);

ALTER TABLE Invoice_Items
ADD CONSTRAINT fk_invoiceitem_billing
FOREIGN KEY (BillID)
REFERENCES Billing (BillID);

ALTER TABLE Payment
ADD CONSTRAINT fk_payment_billing
FOREIGN KEY (BillID)
REFERENCES Billing (BillID);

ALTER TABLE Contracts
ADD CONSTRAINT fk_contract_client
FOREIGN KEY (ClientID)
REFERENCES Clients (ClientID);

ALTER TABLE Purchase_Orders
ADD CONSTRAINT fk_purchaseorder_supplier
FOREIGN KEY (SupplierID)
REFERENCES Suppliers (SupplierID);

ALTER TABLE Expenses
ADD CONSTRAINT fk_expense_employee
FOREIGN KEY (EmployeeID)
REFERENCES Employee (EmployeeID);

ALTER TABLE Resources
ADD CONSTRAINT fk_resource_department
FOREIGN KEY (DepartmentID)
REFERENCES Departments (DepartmentID);

ALTER TABLE Campaigns
ADD CONSTRAINT fk_campaign_project
FOREIGN KEY (ProjectID)
REFERENCES Projects (ProjectID);


ALTER TABLE Strategies
ADD CONSTRAINT fk_strategy_campaign
FOREIGN KEY (CampaignID)
REFERENCES Campaigns (CampaignID);

ALTER TABLE Campaign_Media
ADD CONSTRAINT fk_campaignmedia_campaign
FOREIGN KEY (CampaignID)
REFERENCES Campaigns (CampaignID);

ALTER TABLE Target_Audience
```

```
ADD CONSTRAINT fk_targetaudience_campaign
FOREIGN KEY (CampaignID)
REFERENCES Campaigns (CampaignID);

ALTER TABLE Performance_Metrics
ADD CONSTRAINT fk_performancemetrics_campaign
FOREIGN KEY (CampaignID)
REFERENCES Campaigns (CampaignID);

ALTER TABLE Feedback
ADD CONSTRAINT fk_feedback_campaign
FOREIGN KEY (CampaignID)
REFERENCES Campaigns (CampaignID);

ALTER TABLE ProjectEmployee
ADD CONSTRAINT fk_projectemployee_project
FOREIGN KEY (ProjectID)
REFERENCES Projects (ProjectID);

ALTER TABLE EVENTMEDIACONTACT
ADD CONSTRAINT fk_projectemployee_employee
FOREIGN KEY (EmployeeID)
REFERENCES Employee (EmployeeID);




SELECT
    ca.CampaignID,
    ca.Title AS CampaignTitle,
    pm.MetricID,
    pm.KeyMetric,
    pm.Value,
    pm.DateMeasured
FROM
    Campaigns ca
LEFT JOIN
    Performance_Metrics pm ON ca.CampaignID = pm.CampaignID
ORDER BY
    ca.CampaignID, pm.MetricID;


SELECT
    p.ProjectID,
    p.Title AS ProjectTitle,
    p.Description AS ProjectDescription,
    c.Name AS ClientName,
    COALESCE(SUM(b.Amount), 0) AS TotalBillingAmount,
    COUNT(DISTINCT pe.EmployeeID) AS EmployeeCount,
    (SELECT
        ca.Title
     FROM
        Campaigns ca
     JOIN
```

```
        Performance_Metrics pm ON ca.CampaignID = pm.CampaignID
     WHERE
        ca.ProjectID = p.ProjectID
     ORDER BY
        pm.Value DESC
     FETCH FIRST 1 ROWS ONLY
    ) AS TopCampaign
FROM
    Projects p
JOIN
    Clients c ON p.ClientID = c.ClientID
LEFT JOIN
    Billing b ON p.ProjectID = b.ProjectID
LEFT JOIN
    ProjectEmployee pe ON p.ProjectID = pe.ProjectID
GROUP BY
    p.ProjectID,
    p.Title,
    p.Description,
    c.Name
ORDER BY
    p.ProjectID;


SELECT
    p.ProjectID,
    p.Title AS ProjectTitle,
    p.Description AS ProjectDescription,
    c.Name AS ClientName,
    COALESCE(SUM(b.Amount), 0) AS TotalBillingAmount,
    (SELECT
        ca.Title
     FROM
        Campaigns ca
     JOIN
        Performance_Metrics pm ON ca.CampaignID = pm.CampaignID
     WHERE
        ca.ProjectID = p.ProjectID
     ORDER BY
        pm.Value DESC
     FETCH FIRST 1 ROWS ONLY
    ) AS TopCampaign
FROM
    Projects p
JOIN
    Clients c ON p.ClientID = c.ClientID
LEFT JOIN
    Billing b ON p.ProjectID = b.ProjectID
GROUP BY
    p.ProjectID,
    p.Title,
    p.Description,
    c.Name
ORDER BY
```

```
    p.ProjectID;


WITH ProjectFinancials AS (
    SELECT
        b.ProjectID,
        SUM(b.Amount) AS TotalBillingAmount,
        COALESCE(SUM(p.Amount), 0) AS TotalPaymentsReceived
    FROM
        Billing b
    LEFT JOIN
        Payment p ON b.BillID = p.BillID
    GROUP BY
        b.ProjectID
)
SELECT
    p.ProjectID,
    p.Title AS ProjectTitle,
    c.Name AS ClientName,
    pf.TotalBillingAmount,
    pf.TotalPaymentsReceived,
    COUNT(DISTINCT pe.EmployeeID) AS EmployeeCount
FROM
    Projects p
JOIN
    Clients c ON p.ClientID = c.ClientID
LEFT JOIN
    ProjectFinancials pf ON p.ProjectID = pf.ProjectID
LEFT JOIN
    ProjectEmployee pe ON p.ProjectID = pe.ProjectID
GROUP BY
    p.ProjectID,
    p.Title,
    c.Name,
    pf.TotalBillingAmount,
    pf.TotalPaymentsReceived
ORDER BY
    p.ProjectID;


SELECT p.Title, p.Description, p.StartDate, p.EndDate
FROM Projects p
WHERE p.ClientID = 23;

#List all employees in a particular department:

SELECT e.FirstName, e.LastName, e.Position
FROM Employee e
INNER JOIN Departments d ON e.DepartmentID = d.DepartmentID
WHERE d.Name = 'Digital Marketing';

#clients and associated projects

SELECT
```

```
    p.ProjectID,
    p.Title AS ProjectTitle,
    c.ClientID,
    c.Name AS ClientName
FROM
    Projects p
JOIN
    Clients c ON p.ClientID = c.ClientID
ORDER BY
    p.ProjectID;

#employee count

SELECT
    d.DepartmentID,
    d.Name AS DepartmentName,
    COUNT(e.EmployeeID) AS EmployeeCount
FROM
    Departments d
LEFT JOIN
    Employee e ON d.DepartmentID = e.DepartmentID
GROUP BY
    d.DepartmentID, d.Name
ORDER BY
    d.Name;


#Calculate Total Billing Amount per Client

SELECT
    c.ClientID,
    c.Name AS ClientName,
    SUM(b.Amount) AS TotalBillingAmount
FROM
    Clients c
JOIN
    Billing b ON c.ClientID = b.ClientID
GROUP BY
    c.ClientID, c.Name
ORDER BY
    TotalBillingAmount DESC;


#Find Top 5 Most Expensive Projects

SELECT
    p.ProjectID,
    p.Title AS ProjectTitle,
    SUM(b.Amount) AS ProjectCost
FROM
    Projects p
JOIN
    Billing b ON p.ProjectID = b.ProjectID
GROUP BY
```

```
    p.ProjectID, p.Title
ORDER BY
    ProjectCost DESC
FETCH FIRST 5 ROWS ONLY;


#Find the Most Recent Event for Each Project

SELECT
    p.ProjectID,
    p.Title AS ProjectTitle,
    e.EventID,
    e.Name AS EventName,
    e.Event_Date
FROM
    Projects p
JOIN
    Events e ON p.ProjectID = e.ProjectID
WHERE
    e.Event_Date = (
        SELECT
            MAX(e2.Event_Date)
        FROM
            Events e2
        WHERE
            e2.ProjectID = p.ProjectID
    )
ORDER BY
    p.ProjectID;


#Provide Project Overview with Top Campaign


SELECT
    p.ProjectID,
    p.Title AS ProjectTitle,
    p.Description AS ProjectDescription,
    c.Name AS ClientName,
    COALESCE(SUM(b.Amount), 0) AS TotalBillingAmount,
    COUNT(DISTINCT pe.EmployeeID) AS EmployeeCount,
    (SELECT
        ca.Title
     FROM
        Campaigns ca
     JOIN
        Performance_Metrics pm ON ca.CampaignID = pm.CampaignID
     WHERE
        ca.ProjectID = p.ProjectID
     ORDER BY
        pm.Value DESC
     FETCH FIRST 1 ROWS ONLY
    ) AS TopCampaign
FROM
```

```
    Projects p
JOIN
    Clients c ON p.ClientID = c.ClientID
LEFT JOIN
    Billing b ON p.ProjectID = b.ProjectID
LEFT JOIN
    ProjectEmployee pe ON p.ProjectID = pe.ProjectID
GROUP BY
    p.ProjectID,
    p.Title,
    p.Description,
    c.Name
ORDER BY
    p.ProjectID;


#Provide the financial overview of the projects

WITH ProjectFinancials AS (
    SELECT
        b.ProjectID,
        SUM(b.Amount) AS TotalBillingAmount,
        COALESCE(SUM(p.Amount), 0) AS TotalPaymentsReceived
    FROM
        Billing b
    LEFT JOIN
        Payment p ON b.BillID = p.BillID
    GROUP BY
        b.ProjectID
)
SELECT
    p.ProjectID,
    p.Title AS ProjectTitle,
    c.Name AS ClientName,
    pf.TotalBillingAmount,
    pf.TotalPaymentsReceived,
    COUNT(DISTINCT pe.EmployeeID) AS EmployeeCount
FROM
    Projects p
JOIN
    Clients c ON p.ClientID = c.ClientID
LEFT JOIN
    ProjectFinancials pf ON p.ProjectID = pf.ProjectID
LEFT JOIN
    ProjectEmployee pe ON p.ProjectID = pe.ProjectID
GROUP BY
    p.ProjectID,
    p.Title,
    c.Name,
    pf.TotalBillingAmount,
    pf.TotalPaymentsReceived
ORDER BY
    p.ProjectID;
```

```
 #Track the performance metrics for each campaign.

SELECT
    ca.CampaignID,
    ca.Title AS CampaignTitle,
    pm.MetricID,
    pm.KeyMetric,
    pm.Value,
    pm.DateMeasured
FROM
    Campaigns ca
LEFT JOIN
    Performance_Metrics pm ON ca.CampaignID = pm.CampaignID
ORDER BY
    ca.CampaignID, pm.MetricID;
```