**MIS-IBDA Project**

**Topic : A – Human Resources Information System**

**Submitted By: Group 2**

Bolleni Sai Srujana,

Harshita Motiramani

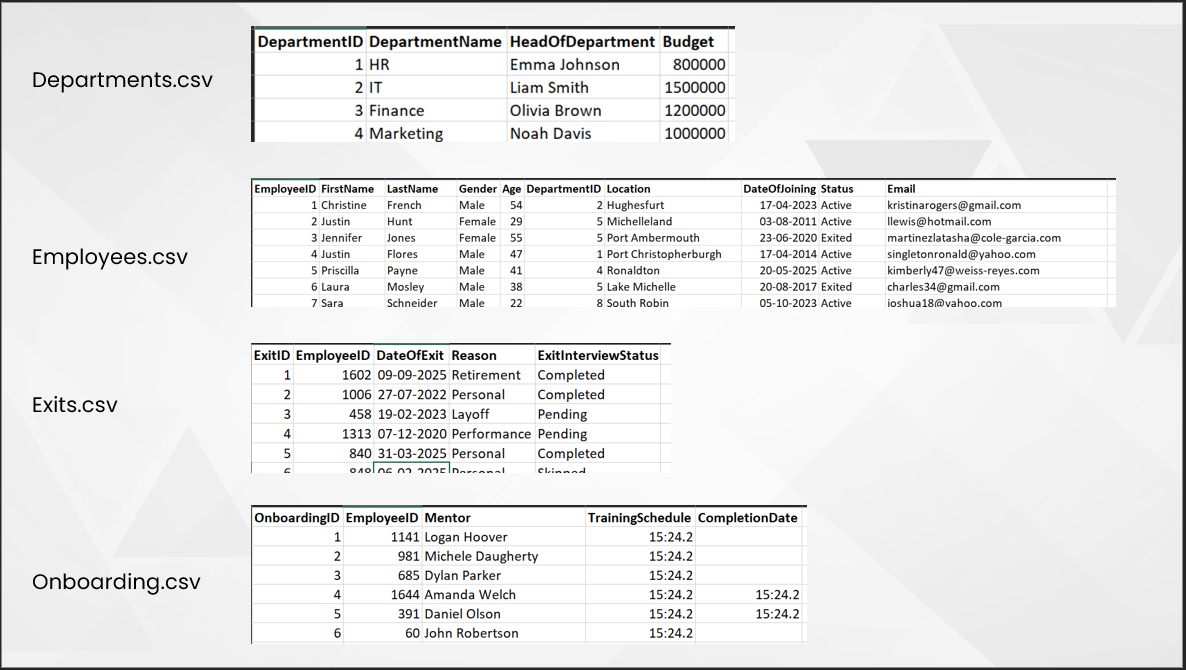
Irish Jain

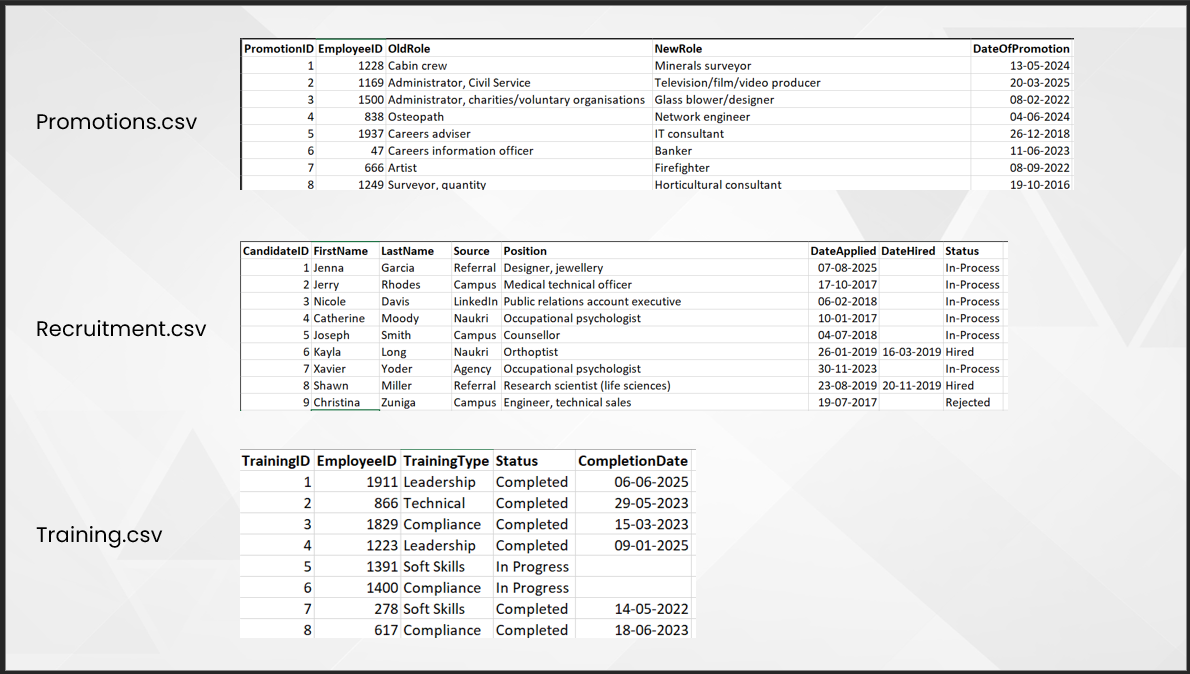
Krishna Dwivedi

Monisha S. Patham

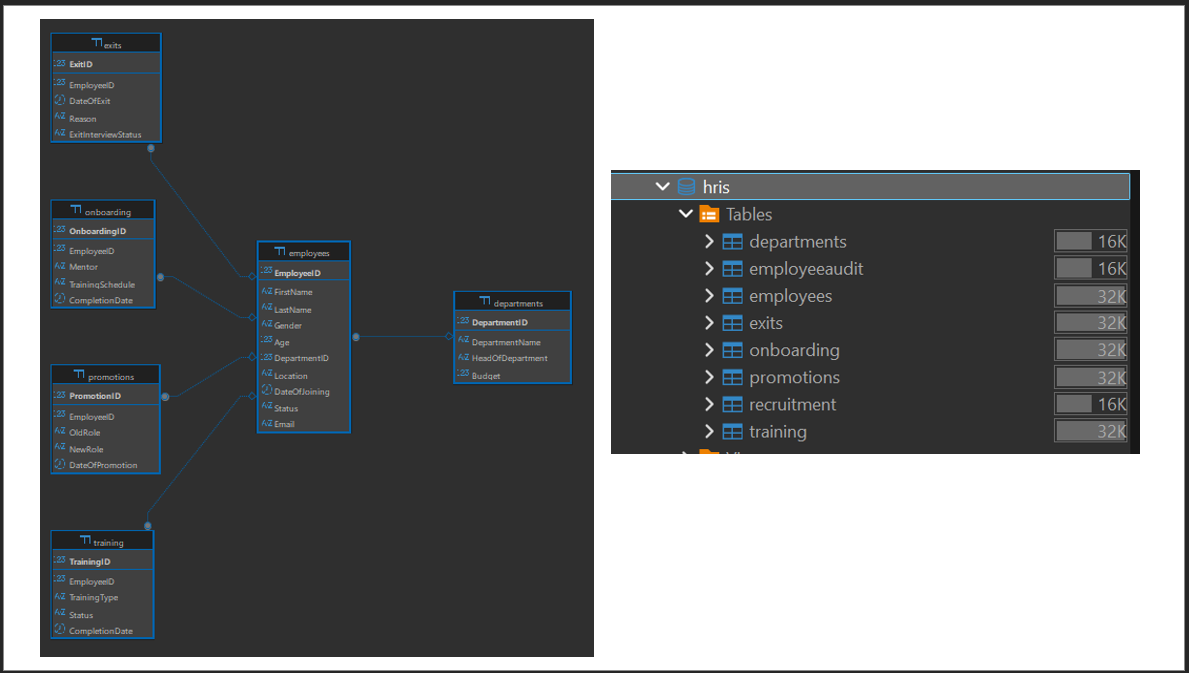
Pallavi Kumari

**Few rows from dummy data**

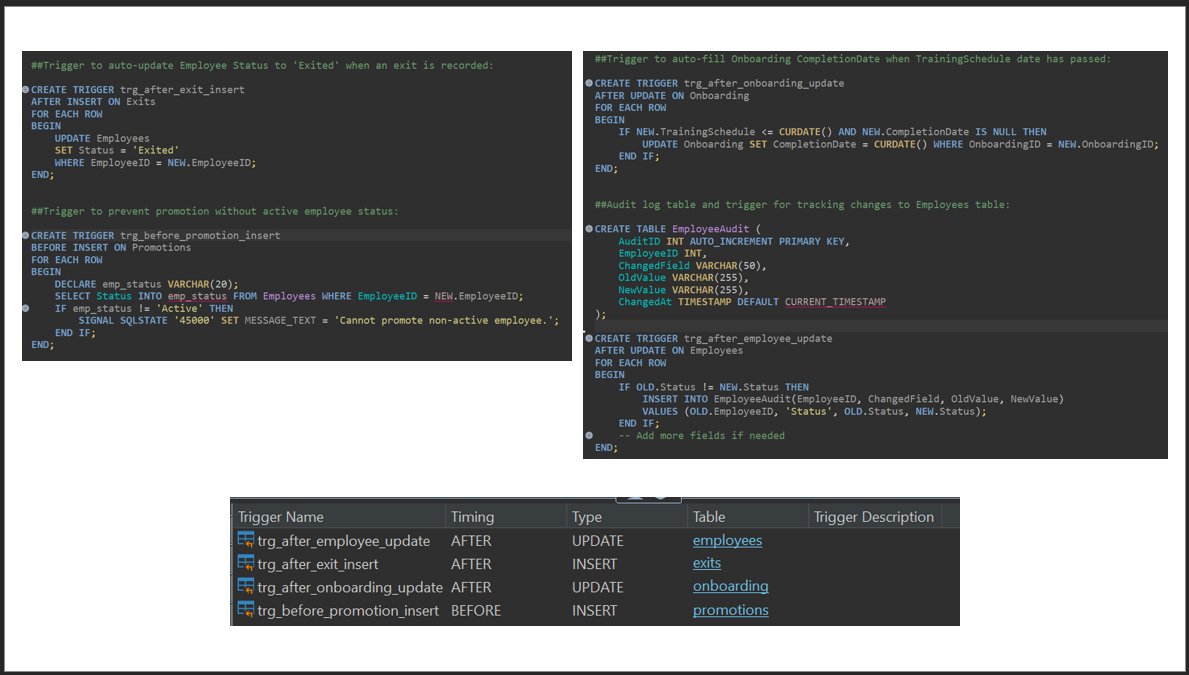




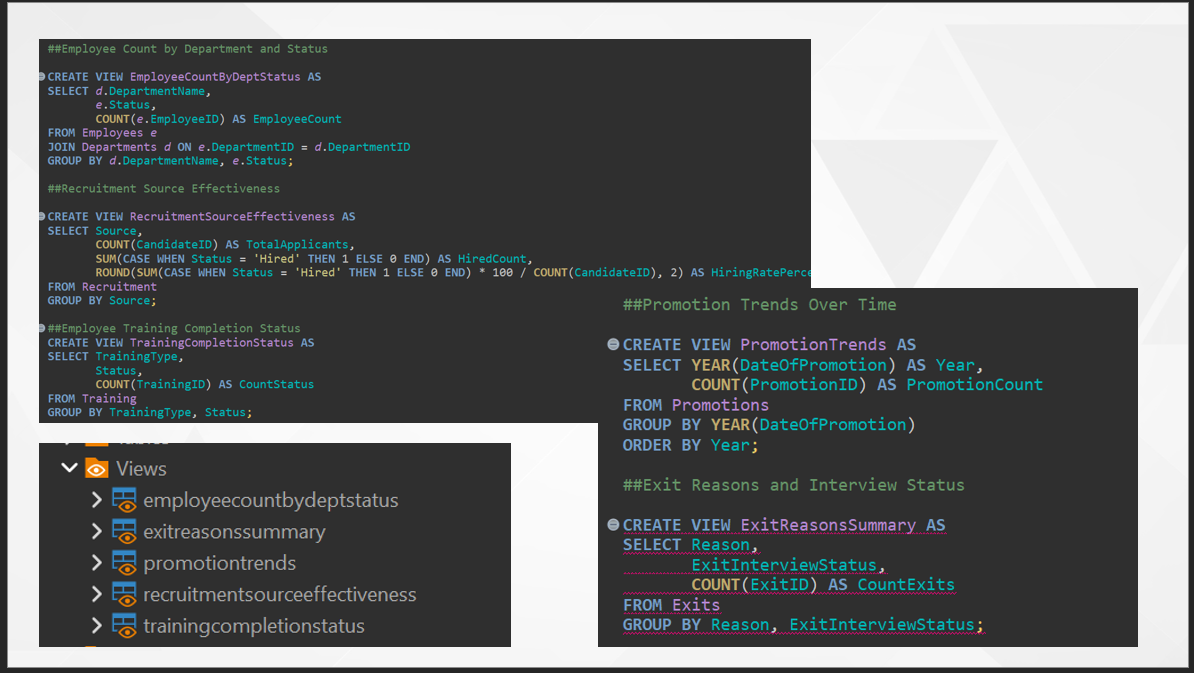
**ER Diagram**

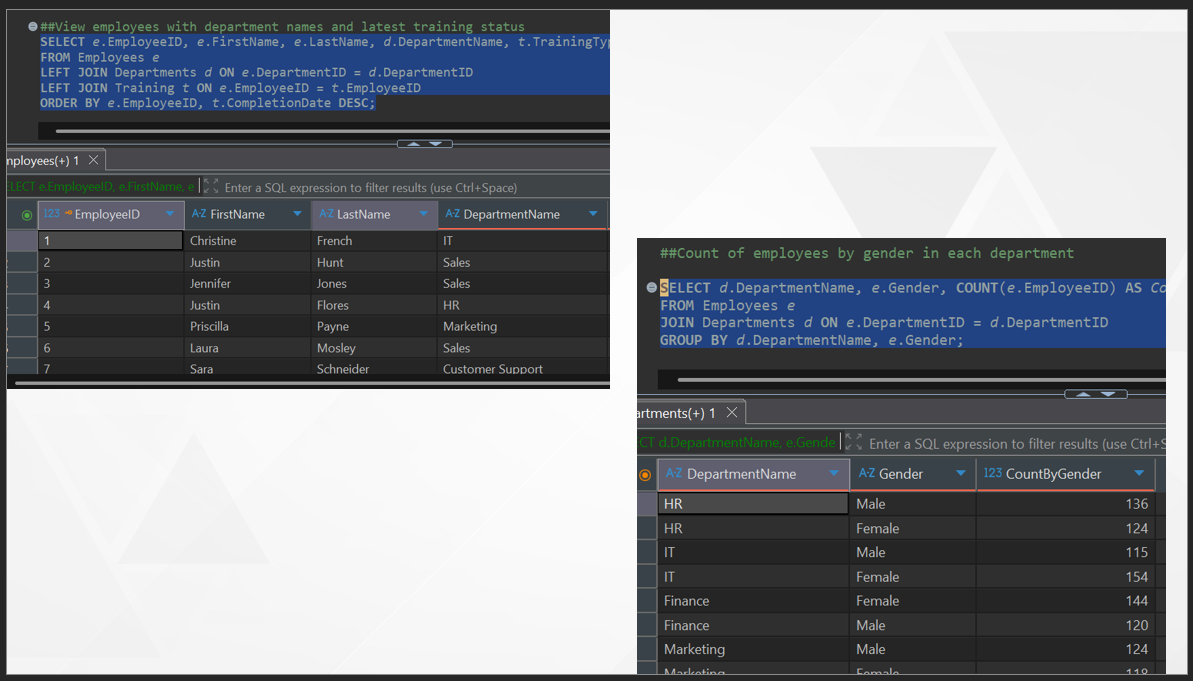


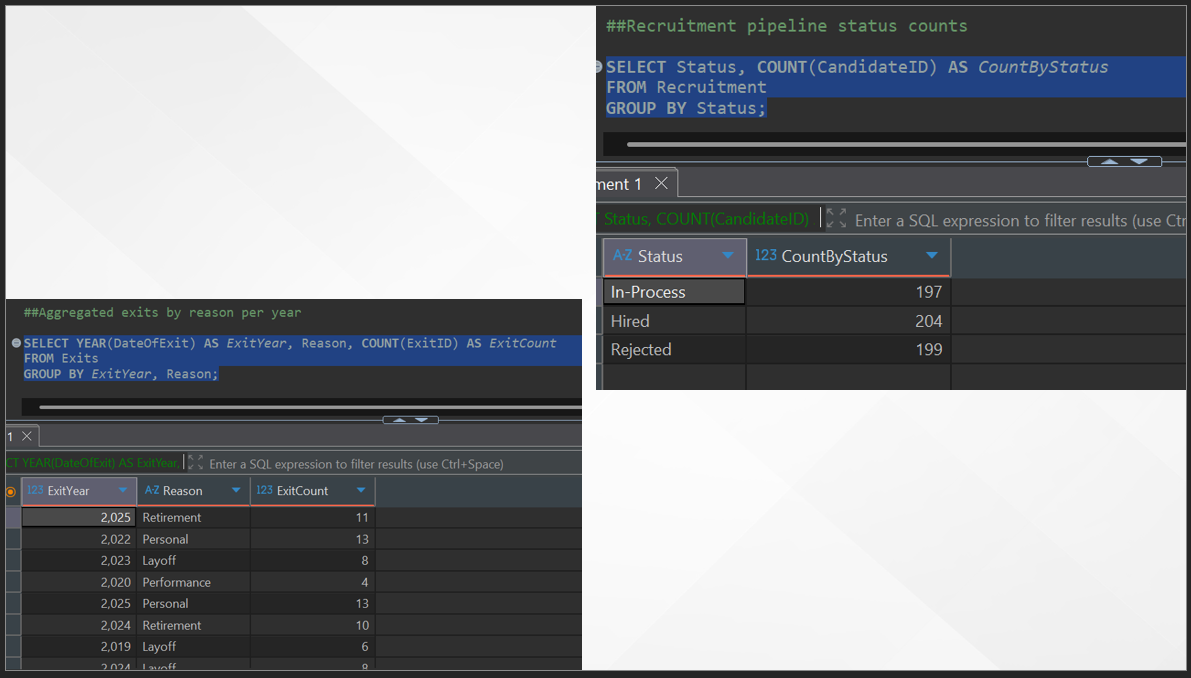
**Triggers**



**Views and Key Queries**





****

**Entire Code**

**CREATE** **TABLE** Departments (

DepartmentID **INT** **PRIMARY** **KEY**,

DepartmentName **VARCHAR**(100) **NOT** **NULL**,

HeadOfDepartment **VARCHAR**(100),

Budget **DECIMAL**(15,2)

);

**CREATE** **TABLE** Employees (

EmployeeID **INT** **PRIMARY** **KEY**,

FirstName **VARCHAR**(50),

LastName **VARCHAR**(50),

Gender **VARCHAR**(10),

Age **INT**,

DepartmentID **INT**,

Location **VARCHAR**(100),

DateOfJoining **DATE**,

Status **VARCHAR**(20),

Email **VARCHAR**(100),

**FOREIGN** **KEY** (DepartmentID) **REFERENCES** Departments(DepartmentID)

);

**CREATE** **TABLE** Training (

TrainingID **INT** **PRIMARY** **KEY**,

EmployeeID **INT**,

TrainingType **VARCHAR**(50),

Status **VARCHAR**(20),

CompletionDate **DATE**,

**FOREIGN** **KEY** (EmployeeID) **REFERENCES** Employees(EmployeeID)

);

**CREATE** **TABLE** Onboarding (

OnboardingID **INT** **PRIMARY** **KEY**,

EmployeeID **INT**,

Mentor **VARCHAR**(100),

TrainingSchedule **VARCHAR**(100),

CompletionDate **DATE**,

**FOREIGN** **KEY** (EmployeeID) **REFERENCES** Employees(EmployeeID)

);

**CREATE** **TABLE** Promotions (

PromotionID **INT** **PRIMARY** **KEY**,

EmployeeID **INT**,

OldRole **VARCHAR**(100),

NewRole **VARCHAR**(100),

DateOfPromotion **DATE**,

**FOREIGN** **KEY** (EmployeeID) **REFERENCES** Employees(EmployeeID)

);

**CREATE** **TABLE** Exits (

ExitID **INT** **PRIMARY** **KEY**,

EmployeeID **INT**,

DateOfExit **DATE**,

Reason **VARCHAR**(50),

ExitInterviewStatus **VARCHAR**(20),

**FOREIGN** **KEY** (EmployeeID) **REFERENCES** Employees(EmployeeID)

);

**CREATE** **TABLE** Recruitment (

CandidateID **INT** **PRIMARY** **KEY**,

FirstName **VARCHAR**(50),

LastName **VARCHAR**(50),

**Source** **VARCHAR**(50),

**Position** **VARCHAR**(100),

DateApplied **DATE**,

DateHired **DATE**,

Status **VARCHAR**(20)

);

##Trigger to auto-update Employee Status to 'Exited' when an exit is recorded:

**CREATE** **TRIGGER** trg\_after\_exit\_insert

**AFTER** **INSERT** **ON** Exits

**FOR** **EACH** **ROW**

**BEGIN**

**UPDATE** Employees

**SET** Status = 'Exited'

**WHERE** EmployeeID = **NEW**.EmployeeID;

**END**;

##Trigger to prevent promotion without active employee status:

**CREATE** **TRIGGER** trg\_before\_promotion\_insert

**BEFORE** **INSERT** **ON** Promotions

**FOR** **EACH** **ROW**

**BEGIN**

**DECLARE** emp\_status **VARCHAR**(20);

**SELECT** Status **INTO** emp\_status **FROM** Employees **WHERE** EmployeeID = **NEW**.EmployeeID;

**IF** emp\_status != 'Active' **THEN**

SIGNAL **SQLSTATE** '45000' **SET** MESSAGE\_TEXT = 'Cannot promote non-active employee.';

**END** **IF**;

**END**;

##Trigger to auto-fill Onboarding CompletionDate when TrainingSchedule date has passed:

**CREATE** **TRIGGER** trg\_after\_onboarding\_update

**AFTER** **UPDATE** **ON** Onboarding

**FOR** **EACH** **ROW**

**BEGIN**

**IF** **NEW**.TrainingSchedule <= CURDATE() **AND** **NEW**.CompletionDate **IS** **NULL** **THEN**

**UPDATE** Onboarding **SET** CompletionDate = CURDATE() **WHERE** OnboardingID = **NEW**.OnboardingID;

**END** **IF**;

**END**;

##Audit log table and trigger for tracking changes to Employees table:

**CREATE** **TABLE** EmployeeAudit (

AuditID **INT** AUTO\_INCREMENT **PRIMARY** **KEY**,

EmployeeID **INT**,

ChangedField **VARCHAR**(50),

OldValue **VARCHAR**(255),

NewValue **VARCHAR**(255),

ChangedAt **TIMESTAMP** **DEFAULT** CURRENT\_TIMESTAMP

);

**CREATE** **TRIGGER** trg\_after\_employee\_update

**AFTER** **UPDATE** **ON** Employees

**FOR** **EACH** **ROW**

**BEGIN**

**IF** **OLD**.Status != **NEW**.Status **THEN**

**INSERT** **INTO** EmployeeAudit(EmployeeID, ChangedField, OldValue, NewValue)

**VALUES** (**OLD**.EmployeeID, 'Status', **OLD**.Status, **NEW**.Status);

**END** **IF**;

-- Add more fields if needed

**END**;

##Employee Count by Department and Status

**CREATE** **VIEW** EmployeeCountByDeptStatus **AS**

**SELECT** d.DepartmentName,

e.Status,

**COUNT**(e.EmployeeID) **AS** EmployeeCount

**FROM** Employees e

**JOIN** Departments d **ON** e.DepartmentID = d.DepartmentID

**GROUP** **BY** d.DepartmentName, e.Status;

##Recruitment Source Effectiveness

**CREATE** **VIEW** RecruitmentSourceEffectiveness **AS**

**SELECT** **Source**,

**COUNT**(CandidateID) **AS** TotalApplicants,

**SUM**(**CASE** **WHEN** Status = 'Hired' **THEN** 1 **ELSE** 0 **END**) **AS** HiredCount,

ROUND(**SUM**(**CASE** **WHEN** Status = 'Hired' **THEN** 1 **ELSE** 0 **END**) \* 100 / **COUNT**(CandidateID), 2) **AS** HiringRatePercent

**FROM** Recruitment

**GROUP** **BY** **Source**;

##Employee Training Completion Status

**CREATE** **VIEW** TrainingCompletionStatus **AS**

**SELECT** TrainingType,

Status,

**COUNT**(TrainingID) **AS** CountStatus

**FROM** Training

**GROUP** **BY** TrainingType, Status;

##Promotion Trends Over Time

**CREATE** **VIEW** PromotionTrends **AS**

**SELECT** **YEAR**(DateOfPromotion) **AS** **Year**,

**COUNT**(PromotionID) **AS** PromotionCount

**FROM** Promotions

**GROUP** **BY** **YEAR**(DateOfPromotion)

**ORDER** **BY** **Year**;

##Exit Reasons and Interview Status

**CREATE** **VIEW** ExitReasonsSummary **AS**

**SELECT** Reason,

ExitInterviewStatus,

**COUNT**(ExitID) **AS** CountExits

**FROM** Exits

**GROUP** **BY** Reason, ExitInterviewStatus;

##View employees with department names and latest training status

**SELECT** e.EmployeeID, e.FirstName, e.LastName, d.DepartmentName, t.TrainingType, t.Status **AS** TrainingStatus

**FROM** Employees e

**LEFT** **JOIN** Departments d **ON** e.DepartmentID = d.DepartmentID

**LEFT** **JOIN** Training t **ON** e.EmployeeID = t.EmployeeID

**ORDER** **BY** e.EmployeeID, t.CompletionDate **DESC**;

##Count of employees by gender in each department

**SELECT** *d*.DepartmentName, *e*.Gender, **COUNT**(*e*.EmployeeID) **AS** *CountByGender*

**FROM** Employees *e*

**JOIN** Departments *d* **ON** *e*.DepartmentID = *d*.DepartmentID

**GROUP** **BY** *d*.DepartmentName, *e*.Gender;

##List active employees with onboarding incomplete

**SELECT** *e*.EmployeeID, *e*.FirstName, *e*.LastName, *o*.Mentor, *o*.TrainingSchedule, *o*.CompletionDate

**FROM** Employees *e*

**JOIN** Onboarding *o* **ON** *e*.EmployeeID = *o*.EmployeeID

**WHERE** *e*.Status = 'Active' **AND** (*o*.CompletionDate **IS** **NULL** **OR** o.CompletionDate > CURDATE());

##Aggregated exits by reason per year

**SELECT** **YEAR**(DateOfExit) **AS** *ExitYear*, Reason, **COUNT**(ExitID) **AS** *ExitCount*

**FROM** Exits

**GROUP** **BY** ExitYear, Reason;

##Recruitment pipeline status counts

**SELECT** Status, **COUNT**(CandidateID) **AS** *CountByStatus*

**FROM** Recruitment

**GROUP** **BY** Status;

##repuuting triiger before promotion

**CREATE** **TRIGGER** trg\_before\_promotion\_insert

**BEFORE** **INSERT** **ON** Promotions

**FOR** **EACH** **ROW**

**BEGIN**

**IF** **NOT** **EXISTS** (**SELECT** 1 **FROM** Employees **WHERE** EmployeeID = **NEW**.EmployeeID **AND** Status = 'Active') **THEN**

SIGNAL **SQLSTATE** '45000' **SET** MESSAGE\_TEXT = 'Cannot promote non-active employee';

**END** **IF**;

**END**;