Table of Contents

PART ONE: DATABASE SYSTEM DESIGN DOCUMENTS

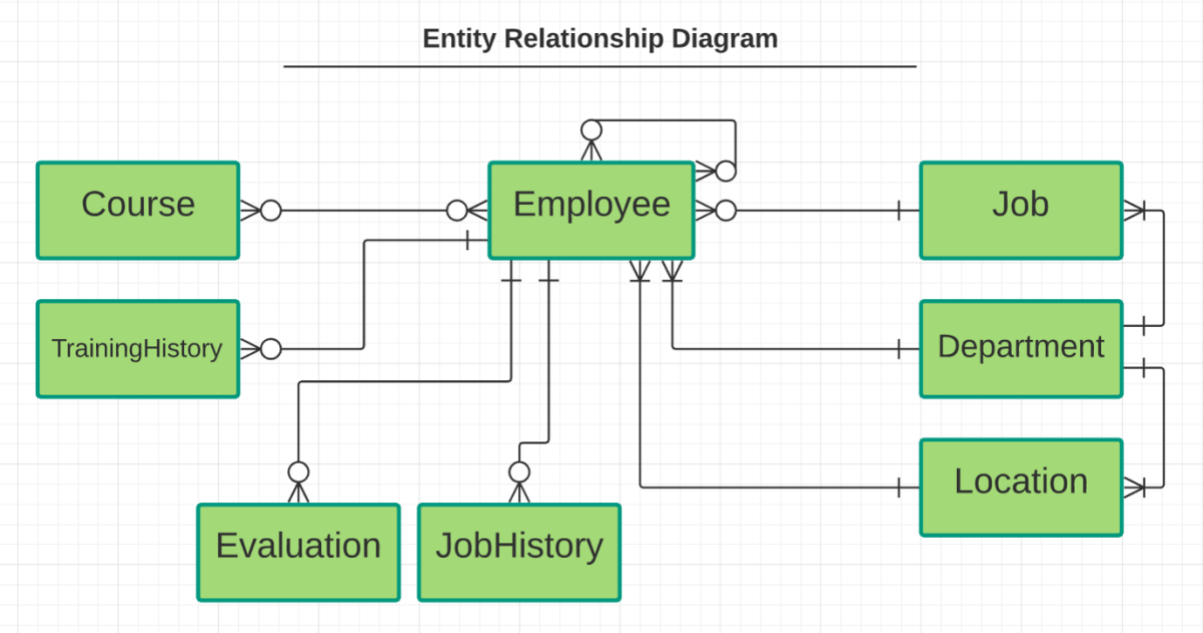
1.1 Map Entity

Entity	Description
Employee	This table includes employees' basic information
Department	This table presents information about different departments in the company
Job	This table contains information about different jobs in different departments
Location	This table includes location information in different areas of the compnay
Course	This table contains information about different training courses provided by the company in different years
Evaluation	This table presents the evaluation information of employees
JobHistory	This is a junction table combining Employee table and Job table with different attributes, which shows the jobs employees has taken
TrainingHistory	This is a junction table combining Employee table and Course table with different attributes, which shows the training courses employees has taken

1.2 Business Rules

Entity	Business Rule
1) Employee	An Employee is a person who works in the Plastic Solutions company
2) Department	A Department is a specialised division of the Plastic Solutions company
3) Job	A Job is the regular work that a person does to earn money
4) Location	A Location is the specified area in the company
5) Course	A Course is set of skill training classes the company provided for employees
6) Evaluation	An Evaluation a reflection of the performance of an employee in the specified year
7) JobHistory	A JobHistory is a record of the previous job an employee took in Plastic Solutions
8) TrainingHistory	A TrainingHistory is a record of the course an employee took
9) Employee and Department	
	a) Each Employee belongs to one Department
	b) Each Department can have many Employees
10) Employee and Job	
	a) Each Employee has one Job in a period of time
	b) Each Job can belong to many Employees
11) Employee and Location	
	a) Each Employee works in one Location
	b) Each Location contains many Employees
12) Employee and Course	
	a) Each Employee can take many Courses
	b) Each Course can be taken by many Employees
13) Employee and Evaluation	
	a) Each Employee can have many Evaluations
	b) Each Evaluation belongs to one Employee
14) Employee and JobHistory	
	a) Each Employee can have many JobHistorys
	b) Each JobHistory corresponds to one Employee
15) Employee and TrainingHistory	
	a) Each Employee can have many TrainingHistorys
	b) Each TrainingHistory corresponds to one Employee
16) Department and Job	
	a) Each Department contains many Jobs
	b) Each Job belongs to one Department
17) Department and Location	
	a) Each Department has many Locations
	b) Each Location corresponds to one Department
18) Employee and Employee	
	a) Employee who is in higher levels can manage many Employees in lower levels
	b) Employee who is in lower levels can be managed by many Employees in higher levels

1.3 Entity Relationship Diagram (ERD)



1.4 Entity Attribute Diagrams (EADs)

1) Employee

EmpID (PK)
FirstName
LastName
Gender
DoB
Age
Email
JobID (FK)
DeptID (FK)
Salary
MonthlyWorkingHours
HireDate
Seniority
RetirementStatus

2) Department

DeptID (PK)
DeptName
EmployeeCount

3) Job

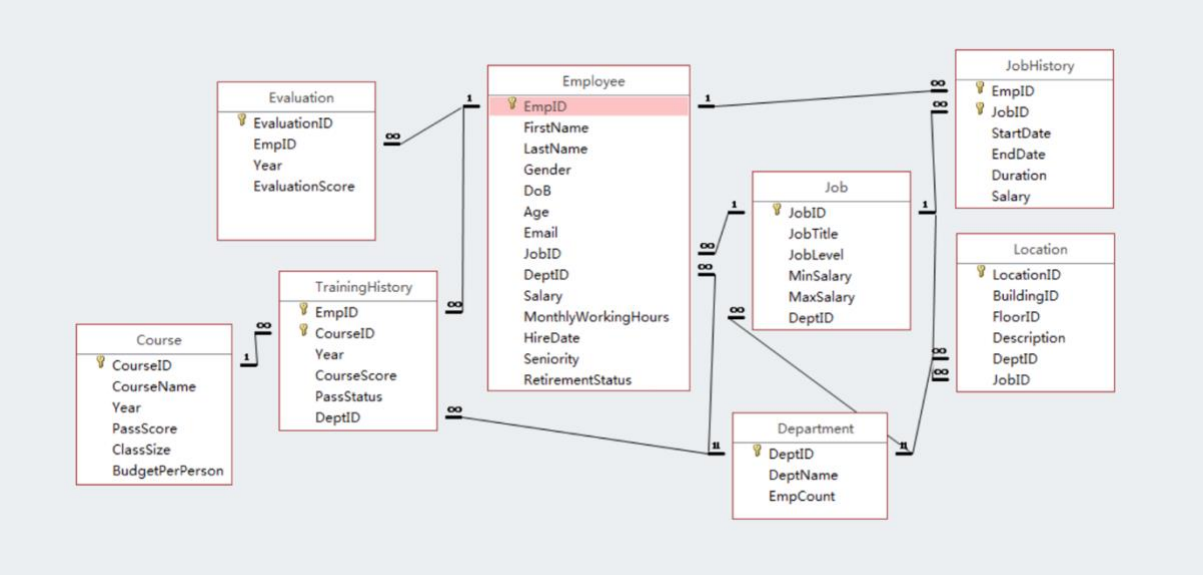
JobID (PK)
JobTitle
JobLevel
MinSalary
MaxSalary
DeptID (FK)

4) Location

LocationID (PK)
BuildingID
FloorID
Description
DeptID (FK)
JobID (FK)

5) Course	6) Evaluation	7) JobHistory	8) TrainingHistory
CourseID (PK)	EvaluationID (PK)	EmpID (PK, FK)	EmpID (PK, FK)
CourseName	EmpID (FK)	JobID (PK, FK)	CourseID (PK, FK)
Year	Year	StartDate	Year
PassScore	EvaluationScore	EndDate	CourseScore
ClassSize		Duration	PassStatus
BudgetPerPerson		Salary	DeptID (FK)
		DeptID (FK)	

1.5 Table Relationship Diagram (TRD)



1.6 Data Schema

Employee {EmpID (PK), FirstName, LastName, Gender, DoB, Age, Email, JobID (FK), DeptID (FK), Salary, MonthlyWorkingHours, HireDate, Seniority, RetirementStatus}

Department {DeptID (PK), DeptName, EmpCount}

Job {JobID (PK), JobTitle, JobLevel, MinSalary, MaxSalary, DeptID (FK)}

Location {LocationID (PK), BuildingID, FloorID, Description, DeptID (FK), JobID (FK)}

Course {CourseID (PK), CourseName, Year, PassScore, ClassSize, BudgetPerPerson}

Evaluation {EvaluationID (PK), EmpID (FK), Year, EvaluationScore}

JobHistory {EmpID (PK, FK), JobID (PK, FK), StartDate, EndDate, Duration, Salary}

TrainingHistory {EmpID (PK, FK), CourseID (PK, FK), Year, CourseScore, PassStatus, DeptID (FK)}

2.7 Data Dictionary

Table	Attribute	Description	Data Type	Range	PK/FK	PK Referenced Table
Employee	EmpID	Unique Employee ID	TEXT	001~999	PK	
	FisrtName	Employee's First Name	TEXT			
	LastName	Employee's Last Name	TEXT			
	Gender	Employee's Gender	TEXT	F, M		
	DoB	Employee's Birthdate	DATE			
	Age	Employee's Age	TEXT			
	Email	Employee's Work Email	TEXT			
	JobID	Unique Job ID	TEXT	01~99	FK	Job
	DeptID	Unique Depatment ID	TEXT	1~9	FK	Department
	Salary	Employee's Current Salary in Pounds	TEXT			
	MonthlyWorkingHours	Employee's Average Monthly Working Hours in 2018	TEXT			
	HireDate	Employee's HireDate	TEXT			
	Seniority	Employee's Service Month in Plastic Solutions	TEXT			
	RetirementStatus	Employee's Retirement Status, 0 for Not Retirement, 1 for Retirement	TEXT	0, 1		
Department	DeptID	Unique Depatment ID	TEXT		PK	
	DeptName	Department's Name	TEXT			
	EmpCount	Number of Employees in this Department	TEXT			
Job	JobID	Unique Job ID	TEXT	01~99	PK	
	JobTitle	Title of Job	TEXT			
	JobLevel	Job Level	TEXT	1~9		
	MinSalary	Minimum Salary for Current Employees who Take This Job	TEXT			
	MaxSalary	Maximum Salary for Current Employees who Take This Job	TEXT			
	DeptID	Unique Depatment ID	TEXT	1~9	FK	Department
Location	LocationID	Unique Location ID	TEXT	10000~99999	PK	
	BuildingID	Unique Building ID	TEXT			
	FloorID	Unique Floor ID in the Building	TEXT			
	Description	Department and Corresponding Job Level in this Location	TEXT			
	DeptID	Unique Depatment ID	TEXT	1~9	FK	Department
	JobID	Unique Job ID	TEXT	01~99	FK	Job
Course	CourseID	Unique Course ID	TEXT	01~99	PK	
	CourseName	Course Name	TEXT			
	Year	Specified Year for This Course	TEXT			
	PassScore	Required Score to Pass This Course	TEXT			
	ClassSize	Optimum Class Size for This Course	TEXT			
	BudgetPerPerson	Course Budget Per Person in Optimum Class Size	TEXT			
Evaluation	EvaluationID	Unique Evaluation ID	TEXT	001~999	PK	
	EmpID	Unique Employee ID	TEXT	001~999	FK	Employee
	Year	Specified Year for This Evaluation	TEXT			
	EvaluationScore	Evaluation Score	TEXT			
JobHistory	EmpID	Unique Employee ID	TEXT	001~999	PK/FK	Employee
	JobID	Unique Job ID	TEXT	01~99	PK/FK	Job
	StartDate	Start Date of This Job	DATE			
	EndDate	End Date of This Job	DATE			
	Duration	Numer of Years Service on This Job	TEXT			
	Salary	Salary Paid for This Job	TEXT			
	DeptID	Unique Depatment ID	TEXT	1~9	FK	Department
TrainingHistory	EmpID	Unique Employee ID	TEXT	001~999	PK/FK	Employee
	CourseID	Unique Course ID	TEXT	01~99	PK/FK	Course
	Year	Specified Year for Taking This Course	TEXT			
	CourseScore	Score for This Course	TEXT	000~100		
	PassStatus	Pass Status, 0 for Not Pass, 1 for Pass	TEXT	0, 1		
	DeptID	Unique Depatment ID	TEXT	1~9	FK	Department

PART TWO: IMPLEMENTATION

SQL Command to Create Department Table

```
1. CREATE TABLE Department (
2. DeptID TEXT PRIMARY KEY,
3. DeptName TEXT,
4. EmpCount TEXT
5. );
```

DeptID	DeptName	EmpCount
1	Human Resources	20
2	Finance	20
3	Sales	30
4	Marketing	30
5	Production	30

SQL Command to Create Job Table

```
1. CREATE TABLE Job (
2. JobID TEXT PRIMARY KEY,
3. JobTitle TEXT,
4. JobLevel TEXT,
5. MinSalary TEXT,
6. MaxSalary TEXT,
7. DeptID TEXT NOT NULL REFERENCES Department (DeptID)
8. );
```

JobID	JobTitle	JobLevel	MinSalary	MaxSalary	DeptID
01	HR Assisant	1	19683	21131	1
02	Human Resources Clerk	2	29853	30879	1
03	Training and Development Coordinator	2	30126	31832	1
04	Payroll Specialist	2	19914	32007	1
05	HR Manager	3	55800	60557	1
06	HR Director	4	82837	89524	1
07	Chief HR Officer	5	212625	212625	1
08	Accountant Clerk	1	21553	23128	2
09	Accountant	2	30955	33013	2
10	Financial Analyst	2	35116	35116	2
11	Senior Accountant	3	67318	71015	2
12	Senior Financial Analyst	3	65692	65891	2
13	Finance Director	4	93181	112993	2
14	Chief Finance Officer	5	222052	222052	2
15	Sales Trainee	1	20356	22535	3
16	Sales Associate	2	29557	33543	3
17	Sales Manager	3	55829	61480	3
18	Sales Director	4	87830	97253	3
19	Chief Sales Officer	5	210360	210360	3
20	Marketing Assistant	1	21552	23361	4
21	Marketing Coordinator	2	30361	32744	4
22	Marketing Manager	3	55622	61309	4
23	Marketing Director	4	86105	108495	4
24	VP of Marketing	5	168600	168600	4
25	Chief Marketing Officer	5	211851	211851	4
26	Production Coordinator	1	19502	21207	5
27	Assistant Producer	1	19809	20978	5
28	Associate Producer	2	28964	32139	5
29	Senior Producer	3	54050	72011	5
30	Producer Manager	4	99286	110460	5
31	Chief Production Officer	5	204818	204818	5

SQL Command to Create Employee Table

```
1.  CREATE TABLE Employee (
2.  EmpID TEXT PRIMARY KEY,
3.  FisrtName TEXT,
4.  LastName TEXT,
5.  Gender TEXT,
6.  DoB DATE,
7.  Age TEXT,
8.  Email TEXT,
9.  JobID TEXT NOT NULL REFERENCES Job (JobID),
10. DeptID TEXT NOT NULL REFERENCES Department (DeptID),
11. Salary TEXT,
12. MonthlyWorkingHours TEXT,
13. HireDate DATE,
14. Seniority TEXT,
15. RetirementStatus TEXT
16. );
```

EmpID	FirstName	LastName	Gender	DoB	Age	Email	JobID	DeptID	Salary	MonthlyWorkingHours	HireDate	Seniority	RetirementStatus
001	Roman	Brewer	M	05/02/1995	23	RomanBrewer@plasticsolutions.com	01	1	19832	184	13/07/2015	3	0
002	Leena	Connor	F	21/03/1995	23	LeenaConnor@plasticsolutions.com	01	1	20082	183	12/03/2016	2	0
003	Kasper	Potts	F	02/09/1991	27	KasperPotts@plasticsolutions.com	01	1	21131	163	24/05/2018	0	0
004	Indi	Mendez	M	12/11/1993	25	IndiMendez@plasticsolutions.com	01	1	19683	206	12/05/2015	3	0
005	Genevieve	Barker	M	17/05/1991	27	GenevieveBarker@plasticsolutions.com	01	1	20464	177	11/11/2014	4	0
006	Brooklyn	Cooper	F	04/06/1993	25	BrooklynCooper@plasticsolutions.com	01	1	20981	190	13/07/2014	4	0
007	Eden	Hartman	M	09/12/1993	25	EdenHartman@plasticsolutions.com	01	1	20034	199	01/07/2016	2	0
008	Annette	Michael	M	03/06/1983	35	AnnetteMichael@plasticsolutions.com	02	1	30879	189	13/11/2011	7	0
009	Lianne	Stamp	F	29/08/1986	32	LianneStamp@plasticsolutions.com	02	1	29853	185	10/09/2011	7	0
010	Herbert	Wood	M	24/12/1989	29	HerbertWood@plasticsolutions.com	03	1	31832	184	18/11/2013	5	0
011	Isabell	Chadwick	F	07/10/1983	35	IsabellChadwick@plasticsolutions.com	03	1	30126	170	13/09/2009	9	0
012	Marguerite	Sanchez	F	01/05/1989	29	MargueriteSanchez@plasticsolutions.com	04	1	29914	205	14/04/2015	3	0
013	Brittney	Bernard	M	11/07/1985	33	BrittneyBernard@plasticsolutions.com	04	1	32007	192	05/09/2007	11	0
014	Jace	Redmond	F	23/08/1974	44	JaceRedmond@plasticsolutions.com	05	1	55800	181	01/04/1999	19	0
015	Elsie	Horner	F	08/09/1976	42	ElsieHorner@plasticsolutions.com	05	1	57083	180	25/05/2001	17	0
016	VIvien	Holmes	M	15/08/1974	44	VIvienHolmes@plasticsolutions.com	05	1	60557	163	17/03/2002	16	0
017	Seamus	Partridge	F	08/08/1969	49	SeamusPartridge@plasticsolutions.com	06	1	88230	147	05/06/1992	26	0
018	Alayah	Denton	F	14/02/1971	47	AlayahDenton@plasticsolutions.com	06	1	82837	183	19/04/1996	22	0
019	Jax	Irvine	M	23/04/1963	55	JaxIrvine@plasticsolutions.com	06	1	89524	148	04/05/1994	24	0
020	Dhruv	Mccray	M	05/08/1959	59	DhruvMccray@plasticsolutions.com	07	1	212625	159	11/07/1985	33	1
021	Ceri	Duarte	F	17/08/1955	63	CeriDuarte@plasticsolutions.com	07	1	220554	135	04/06/1982	34	1
022	Brax	Nicholls	M	15/06/1952	66	BraxNicholls@plasticsolutions.com	07	1	186575	170	15/04/1987	28	1
023	Lester	Gonzales	M	28/04/1991	27	LesterGonzales@plasticsolutions.com	08	2	22333	198	25/05/2017	1	0
024	Olivia	Simmons	F	13/06/1995	23	OliviaSimmons@plasticsolutions.com	08	2	23128	211	29/04/2015	3	0
025	Russell	Reese	M	01/02/1993	25	RussellReese@plasticsolutions.com	08	2	22812	190	01/11/2015	3	0
026	Wilbert	Martinez	M	14/06/1991	27	WilbertMartinez@plasticsolutions.com	08	2	22116	202	21/07/2018	0	0
027	Ella	Hansen	F	06/05/1992	26	EllaHansen@plasticsolutions.com	08	2	22409	175	11/09/2018	0	0
028	Jack	Bass	M	17/08/1993	25	JackBass@plasticsolutions.com	08	2	21553	210	01/05/2015	3	0
029	Francisco	Reid	M	06/09/1993	25	FranciscoReid@plasticsolutions.com	08	2	23073	214	27/08/2018	0	0
030	Belinda	Carter	F	08/04/1984	34	BelindaCarter@plasticsolutions.com	09	2	31388	179	02/11/2015	3	0

SQL Command to Create Location Table

```
1. CREATE TABLE Location (
2. LocationID TEXT PRIMARY KEY,
3. BuildingID TEXT,
4. FloorID TEXT,
5. Description TEXT,
6. DeptID TEXT NOT NULL REFERENCES Department (DeptID),
7. JobID TEXT NOT NULL REFERENCES Job (JobID)
8. );
```

LocationID	BuildingID	FloorID	Description	DeptID	JobID
11101	1	1	HR Level1	1	01
11208	1	1	Finance Level1	2	08
12102	1	2	HR Level2	1	02
12103	1	2	HR Level2	1	03
12104	1	2	HR Level2	1	04
12209	1	2	Finance Level2	2	09
12210	1	2	Finance Level2	2	10
13105	1	3	HR Level3	1	05
13211	1	3	Finance Level3	2	11
13212	1	3	Finance Level3	2	12
21315	2	1	Sales Level1	3	15
22316	2	2	Sales Level2	3	16
22421	2	2	Marketing Level2	4	21
23106	2	3	HR Level4	1	06
23107	2	3	HR Level5	1	07
23213	2	3	Finance Level4	2	13
23214	2	3	Finance Level5	2	14
23318	2	3	Sales Level4	3	18
23319	2	3	Sales Level5	3	19
31420	3	1	Marketing Level1	4	20
31526	3	1	Production Level1	5	26
31527	3	1	Production Level1	5	27
32317	3	2	Sales Level3	3	17
32422	3	2	Marketing Level3	4	22
32528	3	2	Production Level2	5	28
32529	3	2	Production Level3	5	29
33423	3	3	Marketing Level4	4	23
33424	3	3	Marketing Level5	4	24
33425	3	3	Marketing Level5	4	25
33530	3	3	Production Level4	5	30
33531	3	3	Production Level5	5	31

SQL Command to Create Course Table

```
CREATE TABLE Course (
CourseID TEXT PRIMARY KEY,
CourseName TEXT,
Year TEXT,
PassScore TEXT,
ClassSize TEXT,
BudgetPerPerson TEXT
);
```

CourseID	CourseName	Year	PassScore	ClassSize	BudgetPerPerson
10	Writing a good CV	2016	75	15	500
11	Writing a good CV	2017	75	20	500
12	Writing a good CV	2018	75	20	500
20	Writing a good business proposal	2016	70	15	700
21	Writing a good business proposal	2017	70	20	700
22	Writing a good business proposal	2018	70	15	700
30	Fisrt aid training	2017	75	30	850
31	Fisrt aid training	2018	75	35	850
40	Data analytics training	2017	80	25	1200
41	Data analytics training	2018	80	35	1200
50	Communication and presentation skills	2018	75	40	750

SQL Command to Create Evaluation Table

```
1. CREATE TABLE Evaluation (
2. EvaluationID TEXT PRIMARY KEY,
3. EmpID TEXT NOT NULL REFERENCES Employee (EmpID),
4. Year TEXT,
5. EvaluationScore TEXT
6. );
```

EvaluationID	EmpID	Year	EvaluationScore
001	001	2015	70
002	001	2016	68
003	001	2017	74
004	001	2018	61
005	002	2016	82
006	002	2017	72
007	002	2018	62
008	003	2018	80
009	004	2015	69
010	004	2016	80
011	004	2017	66
012	004	2018	69
013	005	2014	81
014	005	2015	68
015	005	2016	65
016	005	2017	68
017	005	2018	93
018	006	2014	61
019	006	2015	77
020	006	2016	60
021	006	2017	76
022	006	2018	84
023	007	2016	75
024	007	2017	88
025	007	2018	61
026	008	2011	77
027	008	2012	68
028	008	2013	74
029	008	2014	78
030	008	2015	86
031	008	2016	89
032	008	2017	87
033	008	2018	73
034	009	2011	85
035	009	2012	82
036	009	2013	72
037	009	2014	71
038	009	2015	68
039	009	2016	61
040	009	2017	62
041	009	2018	64
042	010	2013	74
043	010	2014	72
044	010	2015	69
045	010	2016	69
046	010	2017	77
047	010	2018	88
048	011	2009	60
049	011	2010	75
050	011	2011	84

SQL Command to Create JobHistory Table

```
1.  CREATE TABLE JobHistory (
2.  EmpID TEXT NOT NULL,
3.  JobID TEXT NOT NULL,
4.  StartDate DATE,
5.  EndDate DATE,
6.  Duration TEXT,
7.  Salary TEXT,
8.  PRIMARY KEY (EmpID, JobID),
9.  FOREIGN KEY (EmpID) REFERENCES Employee (EmpID),
10. FOREIGN KEY (JobID) REFERENCES Job (JobID)
11. );
```

EmpID	JobID	StartDate	EndDate	Duration	Salary	DeptID
008	01	13/11/2011	05/07/2014	3	20202	1
009	01	10/09/2011	12/02/2015	4	19603	1
010	01	18/11/2013	21/03/2016	3	19604	1
011	01	13/09/2009	04/05/2014	5	20311	1
012	01	14/04/2015	11/07/2018	3	19573	1
013	01	05/09/2007	12/09/2012	5	20655	1
014	01	01/04/1999	30/05/2003	4	19971	1
014	02	30/05/2003	22/05/2011	8	30313	1
015	01	25/05/2001	13/07/2006	5	19666	1
015	04	13/07/2006	20/03/2013	7	30575	1
016	01	17/03/2002	05/03/2005	3	21386	1
016	04	05/03/2005	21/09/2012	7	31158	1
017	01	05/06/1992	23/05/1996	4	19755	1
017	03	23/05/1996	12/08/2003	7	30153	1
017	05	12/08/2003	22/05/2013	10	55118	1
018	01	19/04/1996	12/07/2000	4	20457	1
018	03	12/07/2000	05/06/2008	8	30801	1
018	05	05/06/2008	18/08/2017	9	57701	1
019	01	04/05/1994	13/02/1997	3	20889	1
019	02	13/02/1997	30/07/2004	7	30930	1
019	05	30/07/2004	12/05/2015	11	59699	1
020	01	11/07/1985	08/08/1988	3	20884	1
020	03	08/08/1988	12/05/1994	6	31037	1
020	05	12/05/1994	23/03/2004	10	65088	1
020	06	23/03/2004	28/09/2013	9	57509	1
021	01	04/06/1982	03/05/1986	4	19791	1
021	02	03/05/1986	12/04/1994	8	31067	1
021	05	12/04/1994	07/02/2002	8	54867	1
021	06	07/02/2002	17/03/2012	10	86544	1
021	07	17/03/2012	06/01/2016	4	161900	1
022	01	15/04/1987	21/08/1990	3	21202	1
022	04	21/08/1990	09/09/1996	6	30719	1
022	05	09/09/1996	08/02/2002	6	61803	1
022	06	08/02/2002	14/03/2010	8	101126	1
022	07	14/03/2010	05/07/2015	5	198451	1
030	08	02/11/2015	12/04/2017	2	22923	2
031	08	01/06/2007	09/02/2011	4	22163	2
032	08	25/07/2009	15/11/2013	4	22037	2
033	08	11/08/2011	04/01/2014	3	23214	2
034	08	15/09/1998	03/06/2002	4	22358	2

SQL Command to Create TrainingHistory Table

```
1. CREATE TABLE TrainingHistory (
2. EmpID TEXT NOT NULL,
3. CourseID TEXT NOT NULL,
4. Year TEXT,
5. CourseScore TEXT,
6. PassStatus TEXT,
7. DeptID TEXT NOT NULL REFERENCES Department (DeptID),
8. PRIMARY KEY (EmpID, CourseID),
9. FOREIGN KEY (EmpID) REFERENCES Employee (EmpID),
10. FOREIGN KEY (CourseID) REFERENCES Course (CourseID)
11. );
```

EmpID	CourseID	Year	CourseScore	PassStatus	DeptID
001	11	2017	84	1	1
001	30	2017	62	0	1
001	50	2018	96	1	1
002	10	2016	93	1	1
002	41	2018	60	0	1
002	50	2018	100	1	1
003	20	2016	88	1	1
004	10	2016	72	0	1
004	22	2018	75	1	1
004	40	2017	71	0	1
004	41	2018	82	1	1
005	50	2018	91	1	1
005	30	2017	73	0	1
005	41	2018	98	1	1
005	50	2018	91	1	1
006	10	2016	67	0	1
006	11	2017	60	0	1
006	40	2017	96	1	1
007	11	2017	84	1	1
007	21	2017	71	1	1
007	30	2017	71	0	1
007	31	2018	77	1	1
007	40	2017	88	1	1

PART THREE: SOLUTIONS TO QUEREIES

3.1 What is our typical retirement age?

Method 1:

Step 1: Select retired employees

```
1. SELECT Employee.EmpID, MAX(YEAR(EndDate)) - YEAR(DoB) AS RetirementAge
2. FROM Employee, JobHistory
3. WHERE Employee.EmpID = JobHistory.EmpID
4. AND RetirementStatus = 1
5. GROUP BY Employee.EmpID;
```

EmpID	RetirementAge
021	61
022	63
043	62
044	63
075	63
076	61
077	60
108	64
109	56
140	62
141	63
142	59

Step 2: Use SQL subqueries to count the number of retired employees in each retirement age and order the results to find the typical retirement age

```
1. SELECT Retirement.RetirementAge, COUNT(*) AS NumberCount
2. FROM (
3. SELECT Employee.EmpID, MAX(YEAR(EndDate)) - YEAR(DoB) AS RetirementAge
4. FROM Employee, JobHistory
5. WHERE Employee.EmpID = JobHistory.EmpID
6. AND RetirementStatus = 1
7. GROUP BY Employee.EmpID
8. ) AS Retirement
9. GROUP BY RetirementAge
10. ORDER BY NumberCount DESC;
```

RetirementAge	NumberCount
63	4
61	2
62	2
60	1
64	1
56	1
59	1

Method 2: Find the mode

```
1.  SELECT RetirementAge AS Typical_RetirementAge
2.  FROM Retirement
3.  GROUP BY RetirementAge
4.  HAVING COUNT(*)
5.         >= ALL(
6.         SELECT COUNT(*)
7.         FROM Retirement
8.         GROUP BY RetirementAge
9.         );
```

Typical_RetirementAge
63

Therefore, the typical retirement age is 63 years old.

3.2 What is the youngest age for promotion?

```
1.  SELECT E.EmpID, YEAR(DoB) AS Birthdate, YEAR(EndDate) AS 1st_Promotion_Year,
    YEAR(EndDate) - YEAR(DoB) AS Age_When_1st_Promotion
2.  FROM Employee AS E, JobHistory AS J
3.  WHERE E.EmpID = J.EmpID
4.  AND J.JobID IN
5.  (01, 08, 15, 20, 26, 27)
6.  ORDER BY Age_When_1st_Promotion
7.  LIMIT 10;
```

EmpID	Birthdate	1st_Promotion_Year	Age_When_1st_Promotion
129	1987	2010	23
035	1979	2002	23
131	1979	2002	23
100	1977	2001	24
066	1982	2006	24
065	1978	2002	24
130	1981	2005	24
036	1977	2001	24
034	1978	2002	24
094	1986	2011	25

Therefore, the youngest age getting promotion is 23 years old.

3.3 Do we have at least one first aider for each of our locations?

First aiders are those who took and passed the course "First aid training". Because employees in different job positions are in different locations, we should first find the number of first aiders corresponding to these first aiders' JobID. Then we would find how many first aiders in each location.

Step 1: Find the number of first aiders corresponding to JobID

```
1.  SELECT JobID, COUNT(*) AS Number_of_First_Aiders
2.  FROM Employee
3.  WHERE EmpID IN (
4.  SELECT EmpID
5.  FROM TrainingHistory
6.  WHERE CourseID IN (30, 31)
7.  AND PassStatus = 1
8.  )
9.  GROUP BY JobID;
```

JobID	Number_of_First_Aiders
01	1
05	1
08	2
09	1
11	2
12	1
15	5
16	2
17	1
20	4
21	3
22	1
26	1
27	1
28	3
29	2

Step 2: Use SQL subqueries to find the number of first aiders in each location

```
1.  SELECT LocationID, Number_of_First_Aiders
2.  FROM Location
3.  LEFT JOIN (
4.  SELECT JobID, COUNT(*) AS Number_of_First_Aiders
5.  FROM Employee
6.  WHERE EmpID IN (
7.  SELECT EmpID
```

```
8.  FROM TrainingHistory
9.  WHERE CourseID IN (30, 31)
10. AND PassStatus = 1
11. )
12. GROUP BY JobID
13. ) AS First_Aiders_JobID
14. ON Location.JobID = First_Aiders_JobID.JobID;
```

LocationID	Number_of_First_Aiders
11101	1
13105	1
11208	2
12209	1
13211	2
13212	1
21315	5
22316	2
32317	1
31420	4
22421	3
32422	1
31526	1
31527	1
32528	3
32529	2
12102	
12103	
12104	
12210	
23106	
23107	
23213	
23214	
23318	
23319	
33423	
33424	
33425	
33530	
33531	

In conclusion, not every location has at least one first aider and 15 out of 31 locations do not have first aiders.

3.4 Whose evaluations are on the decline?

We have at most ten years' evaluation scores for employees who join Plastic Solutions for at least ten years, while some employees only serve in this company for just few years. Therefore, we should classify employees in terms of seniority. To be noticed, employees whose seniority is 1 has two years evaluation scores in 2017 and 2018 respectively. In general, we have nine group in which the employees' seniority is 1 to over 9.

1) Group one: seniority = 1

```
1.  SELECT t1.EmpID, t2.EvaluationScore AS S2017, t1.EvaluationScore AS S2018
2.  FROM (SELECT EmpID, EvaluationScore
3.        FROM Evaluation
4.        WHERE Year = 2018
5.        AND EmpID IN (
6.        SELECT EmpID
7.        FROM Employee
8.        WHERE Seniority = 1)
9.        ) AS t1
10. JOIN (SELECT EmpID, EvaluationScore
11.       FROM Evaluation
12.       WHERE Year = 2017
13.       AND EmpID IN (
14.       SELECT EmpID
15.       FROM Employee
16.        WHERE Seniority = 1)
17.       ) AS t2
18. ON t1.EmpID = t2.EmpID
19. WHERE t1.EvaluationScore < t2.EvaluationScore;
```

EmpID	S2017	S2018
023	75	64
051	78	66
059	69	60
080	74	70
081	78	66
083	79	71
091	86	84
112	93	77
117	86	78

The evaluation scores of employees whose EmpID is 023, 051, 059, 080, 081, 083, 091, 112, 117, are on the decline.

2) Group two: seniority = 2

```
1.  SELECT t1.EmpID, t3.EvaluationScore AS S2016, t2.EvaluationScore AS S2017,
    t1.EvaluationScore AS S2018
2.  FROM (SELECT EmpID, EvaluationScore
3.        FROM Evaluation
4.        WHERE Year = 2018
5.        AND EmpID IN (
6.        SELECT EmpID
7.        FROM Employee
8.        WHERE Seniority = 2)
9.        ) AS t1
10. JOIN (SELECT EmpID, EvaluationScore
11.       FROM Evaluation
12.       WHERE Year = 2017
13.       AND EmpID IN (
14.       SELECT EmpID
15.       FROM Employee
16.       WHERE Seniority = 2)
17.       ) AS t2
18. ON t1.EmpID = t2.EmpID
19. JOIN (SELECT EmpID, EvaluationScore
20.       FROM Evaluation
21.       WHERE Year = 2017
22.       AND EmpID IN (
23.       SELECT EmpID
24.       FROM Employee
25.       WHERE Seniority = 2)
26.       ) AS t3
27. ON t2.EmpID = t3.EmpID
28. WHERE t1.EvaluationScore < t2.EvaluationScore
29. AND   t2.EvaluationScore < t3.EvaluationScore;
```

EmpID	S2016	S2017	S2018

No employee in this group having evaluation scores that are on the decline.

3) Group three: Seniority = 3

```
1.  SELECT t1.EmpID, t4.EvaluationScore AS S2015, t3.EvaluationScore AS S2016,
    t2.EvaluationScore AS S2017, t1.EvaluationScore AS S2018
2.  FROM (SELECT EmpID, EvaluationScore
3.        FROM Evaluation
4.        WHERE Year = 2018
5.        AND EmpID IN (
6.        SELECT EmpID
7.        FROM Employee
8.        WHERE Seniority = 3)
```

```
        ) AS t1
JOIN (SELECT EmpID, EvaluationScore
       FROM Evaluation
       WHERE Year = 2017
       AND EmpID IN (
       SELECT EmpID
       FROM Employee
       WHERE Seniority = 3)
       ) AS t2
ON t1.EmpID = t2.EmpID
JOIN (SELECT EmpID, EvaluationScore
       FROM Evaluation
       WHERE Year = 2016
       AND EmpID IN (
       SELECT EmpID
       FROM Employee
       WHERE Seniority = 3)
       ) AS t3
ON t2.EmpID = t3.EmpID
JOIN (SELECT EmpID, EvaluationScore
       FROM Evaluation
       WHERE Year = 2015
       AND EmpID IN (
       SELECT EmpID
       FROM Employee
       WHERE Seniority = 3)
       ) AS t4
ON t3.EmpID = t4.EmpID
WHERE t1.EvaluationScore < t2.EvaluationScore
AND t2.EvaluationScore < t3.EvaluationScore
AND t3.EvaluationScore < t4.EvaluationScore;
```

EmpID	S2015	S2016	S2017	S2018

No employee in this group having evaluation scores that are on the decline.

4) Group four: Seniority = 4

```
SELECT t1.EmpID, t5.EvaluationScore AS S2014, t4.EvaluationScore AS S2015,
   t3.EvaluationScore AS S2016, t2.EvaluationScore AS S2017,
   t1.EvaluationScore AS S2018
FROM (SELECT EmpID, EvaluationScore
       FROM Evaluation
       WHERE Year = 2018
       AND EmpID IN (
       SELECT EmpID
       FROM Employee
       WHERE Seniority = 4)
       ) AS t1
```

```
JOIN (SELECT EmpID, EvaluationScore
      FROM Evaluation
      WHERE Year = 2017
      AND EmpID IN (
      SELECT EmpID
      FROM Employee
      WHERE Seniority = 4)
      ) AS t2
ON t1.EmpID = t2.EmpID
JOIN (SELECT EmpID, EvaluationScore
      FROM Evaluation
      WHERE Year = 2016
      AND EmpID IN (
      SELECT EmpID
      FROM Employee
      WHERE Seniority = 4)
      ) AS t3
ON t2.EmpID = t3.EmpID
JOIN (SELECT EmpID, EvaluationScore
      FROM Evaluation
      WHERE Year = 2015
      AND EmpID IN (
      SELECT EmpID
      FROM Employee
      WHERE Seniority = 4)
      ) AS t4
ON t3.EmpID = t4.EmpID
JOIN (SELECT EmpID, EvaluationScore
      FROM Evaluation
      WHERE Year = 2014
      AND EmpID IN (
      SELECT EmpID
      FROM Employee
      WHERE Seniority = 4)
      ) AS t5
ON t4.EmpID = t5.EmpID
WHERE t1.EvaluationScore < t2.EvaluationScore
AND t2.EvaluationScore < t3.EvaluationScore
AND t3.EvaluationScore < t4.EvaluationScore
AND t4.EvaluationScore < t5.EvaluationScore;
```

EmpID	S2014	S2015	S2016	S2017	S2018
088	92	84	79	76	61

The evaluation scores of employees whose EmpID is 088 on the decline.

5) Group five: Seniority = 5

```
1.  SELECT t1.EmpID, t6.EvaluationScore AS S2013, t5.EvaluationScore AS S2014,
    t4.EvaluationScore AS S2015, t3.EvaluationScore AS S2016,
    t2.EvaluationScore AS S2017, t1.EvaluationScore AS S2018
2.  FROM (SELECT EmpID, EvaluationScore
3.        FROM Evaluation
4.        WHERE Year = 2018
5.        AND EmpID IN (
6.        SELECT EmpID
7.        FROM Employee
8.        WHERE Seniority = 5)
9.        ) AS t1
10. JOIN (SELECT EmpID, EvaluationScore
11.       FROM Evaluation
12.       WHERE Year = 2017
13.       AND EmpID IN (
14.       SELECT EmpID
15.       FROM Employee
16.       WHERE Seniority = 5)
17.       ) AS t2
18. ON t1.EmpID = t2.EmpID
19. JOIN (SELECT EmpID, EvaluationScore
20.       FROM Evaluation
21.       WHERE Year = 2016
22.       AND EmpID IN (
23.       SELECT EmpID
24.       FROM Employee
25.       WHERE Seniority = 5)
26.       ) AS t3
27. ON t2.EmpID = t3.EmpID
28. JOIN (SELECT EmpID, EvaluationScore
29.       FROM Evaluation
30.       WHERE Year = 2015
31.       AND EmpID IN (
32.       SELECT EmpID
33.       FROM Employee
34.       WHERE Seniority = 5)
35.       ) AS t4
36. ON t3.EmpID = t4.EmpID
37. JOIN (SELECT EmpID, EvaluationScore
38.       FROM Evaluation
39.       WHERE Year = 2014
40.       AND EmpID IN (
41.       SELECT EmpID
42.       FROM Employee
43.       WHERE Seniority = 5)
44.       ) AS t5
45. ON t4.EmpID = t5.EmpID
46. JOIN (SELECT EmpID, EvaluationScore
47.       FROM Evaluation
```

```
48.       WHERE Year = 2013
49.       AND EmpID IN (
50.       SELECT EmpID
51.       FROM Employee
52.       WHERE Seniority = 5)
53.       ) AS t6
54. ON t5.EmpID = t6.EmpID
55. WHERE t1.EvaluationScore < t2.EvaluationScore
56. AND t2.EvaluationScore < t3.EvaluationScore
57. AND t3.EvaluationScore < t4.EvaluationScore
58. AND t4.EvaluationScore < t5.EvaluationScore
59. AND t5.EvaluationScore < t6.EvaluationScore;
```

EmpID	S2013	S2014	S2015	S2016	S2017	S2018

No employee in this group having evaluation scores that are on the decline.

6) Group six: Seniority = 6

```
1.  SELECT t1.EmpID, t7.EvaluationScore AS S2012, t6.EvaluationScore AS S2013,
    t5.EvaluationScore AS S2014, t4.EvaluationScore AS S2015,
    t3.EvaluationScore AS S2016, t2.EvaluationScore AS S2017,
    t1.EvaluationScore AS S2018
2.  FROM (SELECT EmpID, EvaluationScore
3.        FROM Evaluation
4.        WHERE Year = 2018
5.        AND EmpID IN (
6.        SELECT EmpID
7.        FROM Employee
8.        WHERE Seniority = 6)
9.        ) AS t1
10. JOIN (SELECT EmpID, EvaluationScore
11.       FROM Evaluation
12.       WHERE Year = 2017
13.       AND EmpID IN (
14.       SELECT EmpID
15.       FROM Employee
16.       WHERE Seniority = 6)
17.       ) AS t2
18. ON t1.EmpID = t2.EmpID
19. JOIN (SELECT EmpID, EvaluationScore
20.       FROM Evaluation
21.       WHERE Year = 2016
22.       AND EmpID IN (
23.       SELECT EmpID
24.       FROM Employee
25.       WHERE Seniority = 6)
26.        ) AS t3
27. ON t2.EmpID = t3.EmpID
```

```
28. JOIN (SELECT EmpID, EvaluationScore
29.       FROM Evaluation
30.       WHERE Year = 2015
31.       AND EmpID IN (
32.       SELECT EmpID
33.       FROM Employee
34.       WHERE Seniority = 6)
35.       ) AS t4
36. ON t3.EmpID = t4.EmpID
37. JOIN (SELECT EmpID, EvaluationScore
38.       FROM Evaluation
39.       WHERE Year = 2014
40.       AND EmpID IN (
41.       SELECT EmpID
42.       FROM Employee
43.       WHERE Seniority = 6)
44.       ) AS t5
45. ON t4.EmpID = t5.EmpID
46. JOIN (SELECT EmpID, EvaluationScore
47.       FROM Evaluation
48.       WHERE Year = 2013
49.       AND EmpID IN (
50.       SELECT EmpID
51.       FROM Employee
52.       WHERE Seniority = 6)
53.       ) AS t6
54. ON t5.EmpID = t6.EmpID
55. JOIN (SELECT EmpID, EvaluationScore
56.       FROM Evaluation
57.       WHERE Year = 2012
58.       AND EmpID IN (
59.       SELECT EmpID
60.       FROM Employee
61.       WHERE Seniority = 6)
62.       ) AS t7
63. ON t6.EmpID = t7.EmpID
64. WHERE t1.EvaluationScore < t2.EvaluationScore
65. AND t2.EvaluationScore < t3.EvaluationScore
66. AND t3.EvaluationScore < t4.EvaluationScore
67. AND t4.EvaluationScore < t5.EvaluationScore
68. AND t5.EvaluationScore < t6.EvaluationScore
69. AND t6.EvaluationScore < t7.EvaluationScore;
```

EmpID	S2012	S2013	S2014	S2015	S2016	S2017	S2018

No employee in this group having evaluation scores that are on the decline.

7) Group seven: Seniority = 7

```
1.  SELECT t1.EmpID, t8.EvaluationScore AS S2011, t7.EvaluationScore AS S2012,
    t6.EvaluationScore AS S2013, t5.EvaluationScore AS S2014,
    t4.EvaluationScore AS S2015, t3.EvaluationScore AS S2016,
    t2.EvaluationScore AS S2017, t1.EvaluationScore AS S2018
2.  FROM (SELECT EmpID, EvaluationScore
3.        FROM Evaluation
4.        WHERE Year = 2018
5.        AND EmpID IN (
6.        SELECT EmpID
7.        FROM Employee
8.        WHERE Seniority = 7)
9.        ) AS t1
10. JOIN (SELECT EmpID, EvaluationScore
11.       FROM Evaluation
12.       WHERE Year = 2017
13.       AND EmpID IN (
14.       SELECT EmpID
15.       FROM Employee
16.       WHERE Seniority = 7)
17.       ) AS t2
18. ON t1.EmpID = t2.EmpID
19. JOIN (SELECT EmpID, EvaluationScore
20.       FROM Evaluation
21.       WHERE Year = 2016
22.       AND EmpID IN (
23.       SELECT EmpID
24.       FROM Employee
25.       WHERE Seniority = 7)
26.       ) AS t3
27. ON t2.EmpID = t3.EmpID
28. JOIN (SELECT EmpID, EvaluationScore
29.       FROM Evaluation
30.       WHERE Year = 2015
31.       AND EmpID IN (
32.       SELECT EmpID
33.       FROM Employee
34.       WHERE Seniority = 7)
35.       ) AS t4
36. ON t3.EmpID = t4.EmpID
37. JOIN (SELECT EmpID, EvaluationScore
38.       FROM Evaluation
39.       WHERE Year = 2014
40.       AND EmpID IN (
41.       SELECT EmpID
42.       FROM Employee
43.       WHERE Seniority = 7)
44.       ) AS t5
45. ON t4.EmpID = t5.EmpID
46. JOIN (SELECT EmpID, EvaluationScore
47.       FROM Evaluation
48.       WHERE Year = 2013
```

```
49.       AND EmpID IN (
50.       SELECT EmpID
51.       FROM Employee
52.       WHERE Seniority = 7)
53.       ) AS t6
54. ON t5.EmpID = t6.EmpID
55. JOIN (SELECT EmpID, EvaluationScore
56.       FROM Evaluation
57.       WHERE Year = 2012
58.       AND EmpID IN (
59.       SELECT EmpID
60.       FROM Employee
61.       WHERE Seniority = 7)
62.       ) AS t7
63. ON t6.EmpID = t7.EmpID
64. JOIN (SELECT EmpID, EvaluationScore
65.       FROM Evaluation
66.       WHERE Year = 2011
67.       AND EmpID IN (
68.       SELECT EmpID
69.       FROM Employee
70.       WHERE Seniority = 7)
71.       ) AS t8
72. ON t7.EmpID = t8.EmpID
73. WHERE t1.EvaluationScore < t2.EvaluationScore
74. AND t2.EvaluationScore < t3.EvaluationScore
75. AND t3.EvaluationScore < t4.EvaluationScore
76. AND t4.EvaluationScore < t5.EvaluationScore
77. AND t5.EvaluationScore < t6.EvaluationScore
78. AND t6.EvaluationScore < t7.EvaluationScore
79. AND t7.EvaluationScore < t8.EvaluationScore;
```

EmpID	S2011	S2012	S2013	S2014	S2015	S2016	S2017	S2018

No employee in this group having evaluation scores that are on the decline.

8) Group eight: Seniority = 8

```
1.  SELECT t1.EmpID, t9.EvaluationScore AS S2010, t8.EvaluationScore AS S2011, t7.E
    valuationScore AS S2012, t6.EvaluationScore AS S2013, t5.EvaluationScore AS S20
    14, t4.EvaluationScore AS S2015, t3.EvaluationScore AS S2016, t2.EvaluationScor
    e AS S2017, t1.EvaluationScore AS S2018
2.  FROM (SELECT EmpID, EvaluationScore
3.        FROM Evaluation
4.        WHERE Year = 2018
5.        AND EmpID IN (
6.        SELECT EmpID
7.        FROM Employee
```

```
8.        WHERE Seniority = 8)
9.        ) AS t1
10. JOIN (SELECT EmpID, EvaluationScore
11.       FROM Evaluation
12.       WHERE Year = 2017
13.       AND EmpID IN (
14.       SELECT EmpID
15.       FROM Employee
16.       WHERE Seniority = 8)
17.       ) AS t2
18. ON t1.EmpID = t2.EmpID
19. JOIN (SELECT EmpID, EvaluationScore
20.       FROM Evaluation
21.       WHERE Year = 2016
22.       AND EmpID IN (
23.       SELECT EmpID
24.       FROM Employee
25.       WHERE Seniority = 8)
26.       ) AS t3
27. ON t2.EmpID = t3.EmpID
28. JOIN (SELECT EmpID, EvaluationScore
29.       FROM Evaluation
30.       WHERE Year = 2015
31.       AND EmpID IN (
32.       SELECT EmpID
33.       FROM Employee
34.       WHERE Seniority = 8)
35.       ) AS t4
36. ON t3.EmpID = t4.EmpID
37. JOIN (SELECT EmpID, EvaluationScore
38.       FROM Evaluation
39.       WHERE Year = 2014
40.       AND EmpID IN (
41.       SELECT EmpID
42.       FROM Employee
43.       WHERE Seniority = 8)
44.       ) AS t5
45. ON t4.EmpID = t5.EmpID
46. JOIN (SELECT EmpID, EvaluationScore
47.       FROM Evaluation
48.       WHERE Year = 2013
49.       AND EmpID IN (
50.       SELECT EmpID
51.       FROM Employee
52.       WHERE Seniority = 8)
53.       ) AS t6
54. ON t5.EmpID = t6.EmpID
55. JOIN (SELECT EmpID, EvaluationScore
56.       FROM Evaluation
57.       WHERE Year = 2012
58.       AND EmpID IN (
```

```
59.       SELECT EmpID
60.       FROM Employee
61.       WHERE Seniority = 8)
62.       ) AS t7
63. ON t6.EmpID = t7.EmpID
64. JOIN (SELECT EmpID, EvaluationScore
65.       FROM Evaluation
66.       WHERE Year = 2011
67.       AND EmpID IN (
68.       SELECT EmpID
69.       FROM Employee
70.       WHERE Seniority = 8)
71.       ) AS t8
72. ON t7.EmpID = t8.EmpID
73. JOIN (SELECT EmpID, EvaluationScore
74.       FROM Evaluation
75.       WHERE Year = 2010
76.       AND EmpID IN (
77.       SELECT EmpID
78.       FROM Employee
79.       WHERE Seniority = 8)
80.       ) AS t9
81. ON t8.EmpID = t9.EmpID
82. WHERE t1.EvaluationScore < t2.EvaluationScore
83. AND t2.EvaluationScore < t3.EvaluationScore
84. AND t3.EvaluationScore < t4.EvaluationScore
85. AND t4.EvaluationScore < t5.EvaluationScore
86. AND t5.EvaluationScore < t6.EvaluationScore
87. AND t6.EvaluationScore < t7.EvaluationScore
88. AND t7.EvaluationScore < t8.EvaluationScore
89. AND t8.EvaluationScore < t9.EvaluationScore;
```

EmpID	S2010	S2011	S2012	S2013	S2014	S2015	S2016	S2017	S2018

No employee in this group having evaluation scores that are on the decline.

9) Group nine: seniority >= 9

```
1.  SELECT t1.EmpID, t10.EvaluationScore AS S2009,t9.EvaluationScore AS S2010, t8.E
    valuationScore AS S2011, t7.EvaluationScore AS S2012, t6.EvaluationScore AS S20
    13, t5.EvaluationScore AS S2014, t4.EvaluationScore AS S2015, t3.EvaluationScor
    e AS S2016, t2.EvaluationScore AS S2017, t1.EvaluationScore AS S2018
2.  FROM (SELECT EmpID, EvaluationScore
3.        FROM Evaluation
4.        WHERE Year = 2018
5.        AND EmpID IN (
6.        SELECT EmpID
7.        FROM Employee
8.        WHERE Seniority >= 9)
```

```
        ) AS t1
JOIN (SELECT EmpID, EvaluationScore
        FROM Evaluation
        WHERE Year = 2017
        AND EmpID IN (
        SELECT EmpID
        FROM Employee
        WHERE Seniority >= 9)
        ) AS t2
ON t1.EmpID = t2.EmpID
JOIN (SELECT EmpID, EvaluationScore
        FROM Evaluation
        WHERE Year = 2016
        AND EmpID IN (
        SELECT EmpID
        FROM Employee
        WHERE Seniority >= 9)
        ) AS t3
ON t2.EmpID = t3.EmpID
JOIN (SELECT EmpID, EvaluationScore
        FROM Evaluation
        WHERE Year = 2015
        AND EmpID IN (
        SELECT EmpID
        FROM Employee
        WHERE Seniority >= 9)
        ) AS t4
ON t3.EmpID = t4.EmpID
JOIN (SELECT EmpID, EvaluationScore
        FROM Evaluation
        WHERE Year = 2014
        AND EmpID IN (
        SELECT EmpID
        FROM Employee
        WHERE Seniority >= 9)
        ) AS t5
ON t4.EmpID = t5.EmpID
JOIN (SELECT EmpID, EvaluationScore
        FROM Evaluation
        WHERE Year = 2013
        AND EmpID IN (
        SELECT EmpID
        FROM Employee
        WHERE Seniority >= 9)
        ) AS t6
ON t5.EmpID = t6.EmpID
JOIN (SELECT EmpID, EvaluationScore
        FROM Evaluation
        WHERE Year = 2012
        AND EmpID IN (
        SELECT EmpID
```

```
60.       FROM Employee
61.       WHERE Seniority >= 9)
62.       ) AS t7
63. ON t6.EmpID = t7.EmpID
64. JOIN (SELECT EmpID, EvaluationScore
65.       FROM Evaluation
66.       WHERE Year = 2011
67.       AND EmpID IN (
68.       SELECT EmpID
69.       FROM Employee
70.       WHERE Seniority >= 9)
71.       ) AS t8
72. ON t7.EmpID = t8.EmpID
73. JOIN (SELECT EmpID, EvaluationScore
74.       FROM Evaluation
75.       WHERE Year = 2010
76.       AND EmpID IN (
77.       SELECT EmpID
78.       FROM Employee
79.       WHERE Seniority >= 9)
80.       ) AS t9
81. ON t8.EmpID = t9.EmpID
82. JOIN (SELECT EmpID, EvaluationScore
83.       FROM Evaluation
84.       WHERE Year = 2009
85.       AND EmpID IN (
86.       SELECT EmpID
87.       FROM Employee
88.       WHERE Seniority >= 9)
89.       ) AS t10
90. ON t9.EmpID = t10.EmpID
91. WHERE t1.EvaluationScore < t2.EvaluationScore
92. AND t2.EvaluationScore < t3.EvaluationScore
93. AND t3.EvaluationScore < t4.EvaluationScore
94. AND t4.EvaluationScore < t5.EvaluationScore
95. AND t5.EvaluationScore < t6.EvaluationScore
96. AND t6.EvaluationScore < t7.EvaluationScore
97. AND t7.EvaluationScore < t8.EvaluationScore
98. AND t8.EvaluationScore < t9.EvaluationScore
99. AND t9.EvaluationScore < t10.EvaluationScore;
```

EmpID	S2009	S2010	S2011	S2012	S2013	S2014	S2015	S2016	S2017	S2018

No employee in this group having evaluation scores that are on the decline.

In conclusion, the evaluation scores of employees whose EmpID is 023, 051, 059, 080, 081, 083, 091, 112, 117, 088 are on the decline.

3.5 Who should be considered for promotion in 2018?

Different employees have served the company for different years, so they have different seniority. It would take longer time and be harder to get promotion when the employee's seniority is higher. Promotion rules should also be set for different groups of employees in terms of seniority.

The promotion criteria are listed below:

1) For those whose seniority is at most four years, when his or her evaluation score in 2018 is equal to or above 80 and have passed at least two training courses should be considered for promotion in 2018. The employees whose seniority is closer to four years should have higher priority for promotion.
2) For the group of employees whose seniority is between five and ten years, his or her evaluation score is at least 85 and passed at least two courses could be considered for promotion. Also, if the employee's seniority is higher, then the promotion priority should be higher.
3) For employees served over ten years, those who have evaluation score should be at least 90 and have passed at least four courses, is eligible for getting promotion in 2018.

Step 1: Find employees who are eligible for promotion and their EmpID
1) Group one: seniority <= 4

```
SELECT E1.EmpID, Seniority, EvaluationScore
FROM Employee AS E1, Evaluation AS E2
WHERE E2.Year = 2018
AND EvaluationScore >= 80
AND E1.EmpID = E2.EmpID
AND Seniority <= 4
AND E1.EmpID IN (
SELECT EmpID
FROM TrainingHistory
WHERE PassStatus = 1
GROUP BY EmpID
HAVING COUNT(*) >= 2
);
```

EmpID	Seniority	EvaluationScore
005	4	93
078	1	87
079	4	99
082	4	80
084	4	83
110	4	81
119	3	87

Employees whose EmpID is 005, 078, 079, 082, 084, 110 and 119 should be considered for promotion in 2018.

2) Group two: 5 =< seniority =< 10

```
1.  SELECT E1.EmpID, Seniority, EvaluationScore
2.  FROM Employee AS E1, Evaluation AS E2
3.  WHERE E2.Year = 2018
4.  AND EvaluationScore >= 85
5.  AND E1.EmpID = E2.EmpID
6.  AND Seniority BETWEEN 5 AND 10
7.  AND E1.EmpID IN (
8.  SELECT EmpID
9.  FROM TrainingHistory
10. WHERE PassStatus = 1
11. GROUP BY EmpID
12. HAVING COUNT(*) >= 3
13. );
```

EmpID	Seniority	EvaluationScore
063	9	91
093	10	85
123	5	93

Further, analyse how long these employees have stayed in their current position

```
1.  SELECT E.EmpID, Seniority, Seniority - Duration AS Years_Spend_on_Current_Job
2.  FROM Employee AS E, JobHistory AS J
3.  WHERE E.EmpID = J.EmpID
4.  AND E.EmpID IN (063, 093, 123)
5.  ORDER BY Years_Spend_on_Current_Job DESC;
```

EmpID	Seniority	Years_Spend_on_Current_Job
093	10	6
063	9	4
123	5	2

From the above results, we can see employee whose EmpID is 093 spend longest time in current job and is the most senior one. Although with a relatively low evaluation score, this employee should be most considered for promotion.

3) Group three: seniority > 10

```
1.  SELECT E1.EmpID, Seniority, EvaluationScore
2.  FROM Employee AS E1, Evaluation AS E2
3.  WHERE E2.Year = 2018
4.  AND EvaluationScore >= 90
5.  AND E1.EmpID = E2.EmpID
6.  AND Seniority > 10
7.  AND E1.EmpID IN (
8.  SELECT EmpID
9.  FROM TrainingHistory
10. WHERE PassStatus = 1
11. GROUP BY EmpID
12. HAVING COUNT(*) >= 4
13. );
```

EmpID	Seniority	EvaluationScore
068	12	90

Employees whose EmpID is 068 should be considered for promotion in 2018.

Step 2: Get the details of employees who are eligible for promotion in 2018

```
1.  SELECT EmpID, FirstName, LastName, Gender, Age, JobID, DeptID, Seniority
2.  FROM Employee
3.  WHERE EmpID IN (005, 078, 079, 082, 084, 110, 119, 093, 068)
4.  ORDER BY DeptID;
```

EmpID	FirstName	LastName	Gender	Age	JobID	DeptID	Seniority
005	Genevieve	Barker	M	27	01	1	4
068	Norah	Adams	F	36	17	3	12
078	Paula	Aston	F	26	20	4	1
079	Ailbert	Watkins	M	26	20	4	4
082	Buffy	Baxter	F	22	20	4	4
084	Inis	Gardner	M	23	20	4	4
093	Riannon	Farrell	F	33	21	4	10
110	Aileen	Dickenson	F	27	26	5	4
119	Lavinia	Marshell	F	23	27	5	3

The employees listed above should be considered for promotion in 2018. For Paula Aston, her seniority is just one year, so more factors are suggested to be included to evaluate her qualification of promotion.

3.6 Is the training budget being shared fairly among the departments?

Step 1: Get the number of participants for each course in all years since the course is first offered

CV, BP, FA, DA and CP stand for course named "Writing a good CV", "Writing a good business proposal", "First aid training", "Data analytics training" and "Communication and presentation skills" respectively.

```
1. SELECT T.DeptID, D.DeptName, COUNT(*) AS Number_of_Participants_CV
2. FROM TrainingHistory AS T, Department AS D
3. WHERE T.DeptID = D.DeptID
4. AND CourseID IN (10, 11, 12)
5. GROUP BY DeptID;
```

DeptID	DeptName	Number_of_Participants_CV
1	Human Resources	11
2	Finance	15
3	Sales	9
4	Marketing	10
5	Production	15

```
1. SELECT T.DeptID, D.DeptName, COUNT(*) AS Number_of_Participants_BP
2. FROM TrainingHistory AS T, Department AS D
3. WHERE T.DeptID = D.DeptID
4. AND CourseID IN (20, 21, 22)
5. GROUP BY DeptID;
```

DeptID	DeptName	Number_of_Participants_BP
1	Human Resources	6
2	Finance	6
3	Sales	12
4	Marketing	11
5	Production	10

```
1. SELECT T.DeptID, D.DeptName, COUNT(*) AS Number_of_Participants_FA
2. FROM TrainingHistory AS T, Department AS D
3. WHERE T.DeptID = D.DeptID
4. AND CourseID IN (30, 31)
5. GROUP BY DeptID;
```

DeptID	DeptName	Number_of_Participants_FA
1	Human Resources	8
2	Finance	12
3	Sales	16
4	Marketing	17
5	Production	13

```
1. SELECT T.DeptID, D.DeptName, COUNT(*) AS Number_of_Participants_DA
2. FROM TrainingHistory AS T, Department AS D
3. WHERE T.DeptID = D.DeptID
4. AND CourseID IN (40, 41)
5. GROUP BY DeptID;
```

DeptID	DeptName	Number_of_Participants_DA
1	Human Resources	10
2	Finance	9
3	Sales	18
4	Marketing	15
5	Production	12

```
1. SELECT T.DeptID, D.DeptName, COUNT(*) AS Number_of_Participants_CP
2. FROM TrainingHistory AS T, Department AS D
3. WHERE T.DeptID = D.DeptID
4. AND CourseID IN (50)
5. GROUP BY DeptID;
```

DeptID	DeptName	Number_of_Participants_CP
1	Human Resources	7
2	Finance	8
3	Sales	7
4	Marketing	11
5	Production	7

Step 2: Select course budget

```
1. SELECT CourseID, BudgetPerPerson
2. FROM Course;
```

CourseID	BudgetPerPerson
10	500
11	500
12	500
20	700
21	700
22	700
30	850
31	850
40	1200
41	1200
50	750

Step 3: As course budget per person for the same courses offered is the same in different years, then we can calculate the training budget for each department

```
1. SELECT t1.DeptID, t1.DeptName, (Number_of_Participants_CV*500 + Number_of_Parti
   cipants_BP*700 + Number_of_Participants_FA*850 + Number_of_Participants_DA*1200
    + Number_of_Participants_CP*750) AS TrainingBudget
2. FROM
3. (SELECT T.DeptID, D.DeptName, COUNT(*) AS Number_of_Participants_CV
4. FROM TrainingHistory AS T, Department AS D
5. WHERE T.DeptID = D.DeptID
6. AND CourseID IN (10, 11, 12)
7. GROUP BY DeptID
8. ) AS t1,
9. (SELECT T.DeptID, D.DeptName, COUNT(*) AS Number_of_Participants_BP
10. FROM TrainingHistory AS T, Department AS D
11. WHERE T.DeptID = D.DeptID
12. AND CourseID IN (20, 21, 22)
13. GROUP BY DeptID
14. ) AS t2,
15. (SELECT T.DeptID, D.DeptName, COUNT(*) AS Number_of_Participants_FA
16. FROM TrainingHistory AS T, Department AS D
17. WHERE T.DeptID = D.DeptID
18. AND CourseID IN (30, 31)
19. GROUP BY DeptID
20. ) AS t3,
21. (SELECT T.DeptID, D.DeptName, COUNT(*) AS Number_of_Participants_DA
```

```
22. FROM TrainingHistory AS T, Department AS D
23. WHERE T.DeptID = D.DeptID
24. AND CourseID IN (40, 41)
25. GROUP BY DeptID
26. ) AS t4,
27. (SELECT T.DeptID, D.DeptName, COUNT(*) AS Number_of_Participants_CP
28. FROM TrainingHistory AS T, Department AS D
29. WHERE T.DeptID = D.DeptID
30. AND CourseID IN (50)
31. GROUP BY DeptID
32. ) AS t5
33. WHERE t1.DeptID = t2.DeptID
34. AND t2.DeptID = t3.DeptID
35. AND t3.DeptID = t4.DeptID
36. AND t4.DeptID = t5.DeptID;
```

DeptID	DeptName	TrainingBudget
1	Human Resources	33750
2	Finance	38700
3	Sales	53350
4	Marketing	53400
5	Production	45200

We can see the training budget for different departments are quite different, with Human Resources department's lowest £33,750 and Marketing department's higher £53,400. There is a difference of £9,250. Humane Resources and Finance department are in disadvantage in sharing courses' training budget. In contrast, Marketing and Sales department are dominant and Production department is in the middle level. Therefore, the training budget has not been fairly shared among departments.

3.7 Is there any evidence of sexism in our organisation that we should investigate further?

Perspective one: Average salary

1) General average salary

```
1. SELECT Gender, ROUND (AVG(Salary), 0) AS AVG_Salary_General
2. FROM Employee
3. GROUP BY Gender;
```

Gender	AVG_Salary_General
M	71214
F	56235

In general, male's average salary is higher than female's for about (71214 – 56235) / 56235 ≈ 26.6%

2) Average salary for those whose seniority at most 4 years

```
1. SELECT Gender, ROUND (AVG(Salary), 0) AS AVG_Salary_4
2. FROM Employee
3. WHERE Seniority <= 4
4. GROUP BY Gender;
```

Gender	AVG_Salary_4
M	21485
F	22252

For employees whose seniority is at most 4 years, the males' salary is lower than females for about (22252 – 21485) / 22252 ≈ 3.5%

3) Average salary for those whose seniority is between 5 and 10 years

```
1. SELECT Gender, ROUND (AVG(Salary), 0) AS AVG_Salary_5_10
2. FROM Employee
3. WHERE Seniority BETWEEN 5 AND 10
4. GROUP BY Gender;
```

Gender	AVG_Salary_5_10
M	31987
F	34131

For employees whose seniority is between 5 and 10 years, the male's salary is lower than female's for about (34131 – 31987) / 34131 ≈ 6.3%

4) Average salary for those whose seniority is at least 11 years

```
1. SELECT Gender, ROUND (AVG(Salary), 0) AS AVG_Salary_11
2. FROM Employee
3. WHERE Seniority >= 11
4. GROUP BY Gender;
```

Gender	AVG_Salary_11
M	133241
F	97717

For employees whose seniority is at least 11 years, the male's salary is higher than female's for about (133241 – 97717) / 97717 ≈ 36.4%

Therefore, in the perspective of average salary, there is no distinct sex discrimination when employees' seniority is less than 11 years, as the salary difference is less than 10%. However, when employees' seniority is higher than 10 years, there is an evidence of sexism when males get paid for 36.4% higher than female.

Perspective two: Average monthly working hours

1) General average monthly working hours

```
1. SELECT Gender, ROUND (AVG(MonthlyWorkingHours), 0) AS AVG_MonthlyWorkingHours_G
   eneral
2. FROM Employee
3. GROUP BY Gender;
```

Gender	AVG_MonthlyWorkingHours_General
M	184
F	183

In general, working hours are basically no difference between male and female.

2) Average monthly working hours for those whose seniority is at most 4 years

```
1. SELECT Gender, ROUND (AVG(MonthlyWorkingHours), 0) AS AVG_MonthlyWorkingHours_4
2. FROM Employee
3. WHERE Seniority <= 4
4. GROUP BY Gender;
```

Gender	AVG_MonthlyWorkingHours_4
M	190
F	186

For those employees whose seniority is at most 4 years, males averagely work (190 – 186) / 186 ≈ 2.2% longer every month.

3) Average monthly working hours for those whose seniority is between 5 and 10 years

```
1. SELECT Gender, ROUND (AVG(MonthlyWorkingHours), 0) AS AVG_MonthlyWorkingHours_5
   _10
2. FROM Employee
3. WHERE Seniority BETWEEN 5 AND 10
4. GROUP BY Gender;
```

Gender	AVG_MonthlyWorkingHours_5_10
M	194
F	188

For those employees whose seniority is between 5 and 10 years, males work (194 – 188) / 188 ≈ 3.2% longer every month.

4) Average monthly working hours for those whose seniority is at least 11 years

```
1. SELECT Gender, ROUND (AVG(MonthlyWorkingHours), 0) AS AVG_MonthlyWorkingHours_1
   1
2. FROM Employee
3. WHERE Seniority > 10
4. GROUP BY Gender;
```

Gender	AVG_MonthlyWorkingHours_11
M	175
F	178

For those employees whose seniority is at most 4 years, males work (178 – 175) / 178 ≈ 1.7% longer every month.

Therefore, in the perspective of workloads, there is no distinct evidence of sexism as there is no distinct difference in monthly working hours.

Perspective three: Training budget

1) General training budget

```
SELECT t1.Gender, (Number_of_CoursesTaken_CV*500 + Number_of_CoursesTaken_BP*700 + Number_of_CoursesTaken_FA*850 + Number_of_CoursesTaken_DA*1200 + Number_of_CoursesTaken_CP*750) AS TrainingBudget_General
FROM
(SELECT Gender, COUNT(*) AS Number_of_CoursesTaken_CV
FROM TrainingHistory AS T, Employee AS E
WHERE T.EmpID = E.EmpID
AND CourseID IN (10, 11, 12)
GROUP BY Gender
) AS t1,
(SELECT Gender, COUNT(*) AS Number_of_CoursesTaken_BP
FROM TrainingHistory AS T, Employee AS E
WHERE T.EmpID = E.EmpID
AND CourseID IN (20, 21, 22)
GROUP BY Gender
) AS t2,
(SELECT Gender, COUNT(*) AS Number_of_CoursesTaken_FA
FROM TrainingHistory AS T, Employee AS E
WHERE T.EmpID = E.EmpID
AND CourseID IN (30, 31)
GROUP BY Gender
) AS t3,
(SELECT Gender, COUNT(*) AS Number_of_CoursesTaken_DA
FROM TrainingHistory AS T, Employee AS E
WHERE T.EmpID = E.EmpID
AND CourseID IN (40, 41)
GROUP BY Gender
) AS t4,
(SELECT Gender, COUNT(*) AS Number_of_CoursesTaken_CP
FROM TrainingHistory AS T, Employee AS E
WHERE T.EmpID = E.EmpID
AND CourseID = 50
GROUP BY Gender
) AS t5
WHERE t1.Gender = t2.Gender
AND t2.Gender = t3.Gender
AND t3.Gender = t4.Gender
AND t4.Gender = t5.Gender;
```

Gender	TrainingBudget_General
F	121750
M	102650

In general, male get (121750 – 102650) / 121750 ≈ 15.7% lower training budget.

2) Training budget for those whose seniority is at most 4 years

```
1.  SELECT t1.Gender, (Number_of_CoursesTaken_CV*500 + Number_of_CoursesTaken_BP*70
    0 + Number_of_CoursesTaken_FA*850 + Number_of_CoursesTaken_DA*1200 + Number_of_
    CoursesTaken_CP*750) AS TrainingBudget_4
2.  FROM
3.  (SELECT Gender, COUNT(*) AS Number_of_CoursesTaken_CV
4.  FROM TrainingHistory AS T, Employee AS E
5.  WHERE T.EmpID = E.EmpID
6.  AND CourseID IN (10, 11, 12)
7.  AND Seniority <= 4
8.  GROUP BY Gender
9.  ) AS t1,
10. (SELECT Gender, COUNT(*) AS Number_of_CoursesTaken_BP
11. FROM TrainingHistory AS T, Employee AS E
12. WHERE T.EmpID = E.EmpID
13. AND CourseID IN (20, 21, 22)
14. AND Seniority <= 4
15. GROUP BY Gender
16. ) AS t2,
17. (SELECT Gender, COUNT(*) AS Number_of_CoursesTaken_FA
18. FROM TrainingHistory AS T, Employee AS E
19. WHERE T.EmpID = E.EmpID
20. AND CourseID IN (30, 31)
21. AND Seniority <= 4
22. GROUP BY Gender
23. ) AS t3,
24. (SELECT Gender, COUNT(*) AS Number_of_CoursesTaken_DA
25. FROM TrainingHistory AS T, Employee AS E
26. WHERE T.EmpID = E.EmpID
27. AND CourseID IN (40, 41)
28. AND Seniority <= 4
29. GROUP BY Gender
30. ) AS t4,
31. (SELECT Gender, COUNT(*) AS Number_of_CoursesTaken_CP
32. FROM TrainingHistory AS T, Employee AS E
33. WHERE T.EmpID = E.EmpID
34. AND CourseID = 50
35. AND Seniority <= 4
36. GROUP BY Gender
37. ) AS t5
38. WHERE t1.Gender = t2.Gender
39. AND t2.Gender = t3.Gender
```

```
40. AND t3.Gender = t4.Gender
41. AND t4.Gender = t5.Gender;
```

Gender	TrainingBudget_4
F	66150
M	56700

For those employees whose seniority is at most 4 years, males receive (66150 - 56700) / 66150 ≈ 14.3% lower training budget.

3) Training budget for those whose seniority is between 5 and 10 years

```
1.  SELECT t1.Gender, (Number_of_CoursesTaken_CV*500 + Number_of_CoursesTaken_BP*70
    0 + Number_of_CoursesTaken_FA*850 + Number_of_CoursesTaken_DA*1200 + Number_of_
    CoursesTaken_CP*750) AS TrainingBudget_5_10
2.  FROM
3.  (SELECT Gender, COUNT(*) AS Number_of_CoursesTaken_CV
4.  FROM TrainingHistory AS T, Employee AS E
5.  WHERE T.EmpID = E.EmpID
6.  AND CourseID IN (10, 11, 12)
7.  AND Seniority BETWEEN 5 AND 10
8.  GROUP BY Gender
9.  ) AS t1,
10. (SELECT Gender, COUNT(*) AS Number_of_CoursesTaken_BP
11. FROM TrainingHistory AS T, Employee AS E
12. WHERE T.EmpID = E.EmpID
13. AND CourseID IN (20, 21, 22)
14. AND Seniority BETWEEN 5 AND 10
15. GROUP BY Gender
16. ) AS t2,
17. (SELECT Gender, COUNT(*) AS Number_of_CoursesTaken_FA
18. FROM TrainingHistory AS T, Employee AS E
19. WHERE T.EmpID = E.EmpID
20. AND CourseID IN (30, 31)
21. AND Seniority BETWEEN 5 AND 10
22. GROUP BY Gender
23. ) AS t3,
24. (SELECT Gender, COUNT(*) AS Number_of_CoursesTaken_DA
25. FROM TrainingHistory AS T, Employee AS E
26. WHERE T.EmpID = E.EmpID
27. AND CourseID IN (40, 41)
28. AND Seniority BETWEEN 5 AND 10
29. GROUP BY Gender
30. ) AS t4,
31. (SELECT Gender, COUNT(*) AS Number_of_CoursesTaken_CP
32. FROM TrainingHistory AS T, Employee AS E
33. WHERE T.EmpID = E.EmpID
34. AND CourseID = 50
```

```
35. AND Seniority BETWEEN 5 AND 10
36. GROUP BY Gender
37. ) AS t5
38. WHERE t1.Gender = t2.Gender
39. AND t2.Gender = t3.Gender
40. AND t3.Gender = t4.Gender
41. AND t4.Gender = t5.Gender;
```

Gender	TrainingBudget_5_10
F	23550
M	25950

For employees whose seniority is between 5 and 10 years, males receive (25950 – 23550) / 23550 ≈ 10.2% higher training budget.

4) Training budget for those whose seniority is at least 11 years

```
1.  SELECT t1.Gender, (Number_of_CoursesTaken_CV*500 + Number_of_CoursesTaken_BP*70
    0 + Number_of_CoursesTaken_FA*850 + Number_of_CoursesTaken_DA*1200 + Number_of_
    CoursesTaken_CP*750) AS TrainingBudget_11
2.  FROM
3.  (SELECT Gender, COUNT(*) AS Number_of_CoursesTaken_CV
4.  FROM TrainingHistory AS T, Employee AS E
5.  WHERE T.EmpID = E.EmpID
6.  AND CourseID IN (10, 11, 12)
7.  AND Seniority >= 11
8.  GROUP BY Gender
9.  ) AS t1,
10. (SELECT Gender, COUNT(*) AS Number_of_CoursesTaken_BP
11. FROM TrainingHistory AS T, Employee AS E
12. WHERE T.EmpID = E.EmpID
13. AND CourseID IN (20, 21, 22)
14. AND Seniority >= 11
15. GROUP BY Gender
16. ) AS t2,
17. (SELECT Gender, COUNT(*) AS Number_of_CoursesTaken_FA
18. FROM TrainingHistory AS T, Employee AS E
19. WHERE T.EmpID = E.EmpID
20. AND CourseID IN (30, 31)
21. AND Seniority >= 11
22. GROUP BY Gender
23. ) AS t3,
24. (SELECT Gender, COUNT(*) AS Number_of_CoursesTaken_DA
25. FROM TrainingHistory AS T, Employee AS E
26. WHERE T.EmpID = E.EmpID
27. AND CourseID IN (40, 41)
28. AND Seniority >= 11
29. GROUP BY Gender
```

```
30. ) AS t4,
31. (SELECT Gender, COUNT(*) AS Number_of_CoursesTaken_CP
32. FROM TrainingHistory AS T, Employee AS E
33. WHERE T.EmpID = E.EmpID
34. AND CourseID = 50
35. AND Seniority >= 11
36. GROUP BY Gender
37. ) AS t5
38. WHERE t1.Gender = t2.Gender
39. AND t2.Gender = t3.Gender
40. AND t3.Gender = t4.Gender
41. AND t4.Gender = t5.Gender;
```

Gender	TrainingBudget_11
F	32050
M	20000

For employees whose seniority is at least 11 years, males receive (32050 – 20000) / 32050 ≈ 37.6% lower training budget.

Therefore, in the perspective of training budget, males are disadvantaged when they receive lower training budget for most cases.

3.8 What is a 'typical' Plastic Solution career?

A typical career should indicate how many years should an employee on average spend on different level's jobs and the possibility of being promoted to different positions in the same level.

1) HR department:

Step 1: Find average years spent in job positions in HR department

```
1. SELECT JobID, ROUND (AVG(Duration), 1) AS AVG_Duration
2. FROM JobHistory
3. WHERE DeptID = 1
4. GROUP BY JobID;
```

JobID	AVG_Duration
01	3.7
02	7.7
03	6.7
04	7.0
05	9.0
06	9.0
07	4.5

Step 2: Select the job details of HR department

```
1. SELECT JobID, JobTitle, JobLevel
2. FROM Job
3. WHERE DeptID = 1;
```

JobID	JobTitle	JobLevel
01	HR Assisant	1
02	Human Resources Clerk	2
03	Training and Development Coordinator	2
04	Payroll Specialist	2
05	HR Manager	3
06	HR Director	4
07	Chief HR Officer	5

Step 3: Find number of people whose JobID was 02, 03 and 04, which are both level 2 positions

```
1. SELECT JobID, Count(*) AS Number_of_People
2. FROM JobHistory
3. WHERE JobID IN (02, 03, 04)
4. GROUP BY JobID;
```

JobID	Number_of_People
02	3
03	3
04	3

We find that level 2 positions are taken by the same possibility (3/9 ≈ 33.3%).

Based on the results of these two steps, we can conclude the typical career path of HR department is:

JobTitle (JobID)	Job Level	Average Years for Promotion
HR Assistant (01)	1	3.7
Human Resources Clerk (02)	2	7.7 (33.3%)
Training and Development Coordinator (03)	2	7.0 (33.3%)
Payroll Specialist (04)	2	6.7 (33.3%)
HR Manager (05)	3	9.0
HR Director (06)	4	9.0
Chief HR Officer (07)	5	4.5 (Years spent on the position)

The percentage in the parenthesis stands for the possibility for taking the same level jobs. For example, there is a 33.3% a HR Assistant would be promoted to Human Resources Clerk, and after promotion normally he or she would spend 7.7 years on the position before next promotion.

2) Finance department:

Step 1: Find average years spent in job positions in Finance department

```
1. SELECT JobID, ROUND (AVG(Duration), 1) AS AVG_Duration
2. FROM JobHistory
3. WHERE DeptID = 2
4. GROUP BY JobID;
```

JobID	AVG_Duration
08	3.5
09	8.0
10	7.0
11	9.0
12	7.2
13	7.3
14	5.5

Step 2: Select the job details of Finance department

```
1. SELECT JobID, JobTitle, JobLevel
2. FROM Job
3. WHERE DeptID = 2;
```

JobID	JobTitle	JobLevel
08	Accountant Clerk	1
09	Accountant	2
10	Financial Analyst	2
11	Senior Accountant	3
12	Senior Financial Analyst	3
13	Finance Director	4
14	Chief Finance Officer	5

Step 3: Find number of people whose JobID was 09 and 10, 11 and 12, which are both at the same level respectively

```
1. SELECT JobID, Count(*) AS Number_of_People
2. FROM JobHistory
3. WHERE JobID IN (09, 10, 11, 12)
4. GROUP BY JobID;
```

JobID	Number_of_People
09	5
10	6
11	3
12	4

For positions in job level 2, there is a 5 / (5+6) ≈ 45.5% possibility for being promoted to the job which JobID = 09, and 54.5% for JobID = 10. For positions

in job level 3, there is a 3 / (3+4) ≈ 42.9% possibility for being promoted to the job which JobID = 11, and 57.1% possibility for JobID = 12.

Therefore, the typical career path for Finance department is:

JobTitle (JobID)	Job Level	Average Years for Promotion
Accountant Clerk (08)	1	3.5
Accountant (09)	2	8.0 (45.5%)
Financial Analyst (10)	2	7.0 (54.5%)
Senior Accountant (11)	3	9.0 (42.9%)
Senior Financial Analyst (12)	3	7.2 (57.1%)
Finance Director (13)	4	7.3
Chief Finance Officer (14)	5	5.5 (Years spent on the position)

3) Sales department:

Step 1: Find average years spent in job positions in Sales department

```
1. SELECT JobID, ROUND (AVG(Duration), 1) AS AVG_Duration
2. FROM JobHistory
3. WHERE DeptID = 3
4. GROUP BY JobID;
```

JobID	AVG_Duration
15	3.6
16	6.7
17	8.8
18	9.8
19	6.7

Step 2: Select the job details of Sales department

```
1. SELECT JobID, JobTitle, JobLevel
2. FROM Job
3. WHERE DeptID = 3;
```

JobID	JobTitle	JobLevel
15	Sales Trainee	1
16	Sales Associate	2
17	Sales Manager	3
18	Sales Director	4
19	Chief Sales Officer	5

Therefore, the typical career path for Sales department is:

JobTitle (JobID)	Job Level	Average Years for Promotion
Sales Trainee (15)	1	3.6
Sales Associate (16)	2	6.7
Sales Manager (17)	3	8.8
Sales Director (18)	4	9.8
Chief Sales Officer (19)	5	6.7 (Years spent on the position)

4) Marketing department:

Step 1: Find average years spent in job positions in Marketing department

```
1. SELECT JobID, ROUND (AVG(Duration), 1) AS AVG_Duration
2. FROM JobHistory
3. WHERE DeptID = 4
4. GROUP BY JobID;
```

JobID	AVG_Duration
20	3.3
21	6.2
22	7.0
23	7.8
24	5.0
25	5.5

Step 2: Select the job details of Marketing department

```
1. SELECT JobID, JobTitle, JobLevel
2. FROM Job
3. WHERE DeptID = 4;
```

JobID	JobTitle	JobLevel
20	Marketing Assistant	1
21	Marketing Coordinator	2
22	Marketing Manager	3
23	Marketing Director	4
24	VP of Marketing	5
25	Chief Marketing Officer	5

Therefore, the typical career path for Marketing department is:

JobTitle (JobID)	Job Level	Average Years for Promotion
Marketing Assistant (20)	1	3.3
Marketing Coordinator (21)	2	6.2
Marketing Manager (22)	3	7.0
Marketing Director (23)	4	7.8
VP of Marketing (24)	5	5.0
Chief Marketing Officer (25)	5	5.5 (Years spent on the position)

5) Production department:

Step 1: Find average years spent in job positions in Production department

```
1. SELECT JobID, ROUND (AVG(Duration), 1) AS AVG_Duration
2. FROM JobHistory
3. WHERE DeptID = 5
4. GROUP BY JobID;
```

JobID	AVG_Duration
26	2.9
27	3.9
28	6.8
29	7.6
30	8.5
31	9.3

Step 2: Select the job details of Production department

```
1. SELECT JobID, JobTitle, JobLevel
2. FROM Job
3. WHERE DeptID = 5;
```

JobID	JobTitle	JobLevel
26	Production Coordinator	1
27	Assistant Producer	1
28	Associate Producer	2
29	Senior Producer	3
30	Producer Manager	4
31	Chief Production Officer	5

Step 3: Find number of people whose JobID was 26 and 27, which are both at the same level

```
1. SELECT JobID, Count(*) AS Number_of_People
2. FROM JobHistory
3. WHERE JobID IN (26, 27)
4. GROUP BY JobID;
```

JobID	Number_of_People
26	10
27	10

We find that there are 10 / 20 = 50% possibility an employee at level 1 is a Production Coordinator and 50% is an Assistant Producer.

Therefore, the typical career path for Production department is:

JobTitle (JobID)	Job Level	Average Years for Promotion
Production Coordinator (26)	1	2.9 (50%)
Assistant Producer (27)	1	3.9 (50%)
Associate Producer (28)	2	6.8
Senior Producer (29)	3	7.6
Producer Manager (30)	4	8.5
Chief Production Officer (31)	5	9.3 (Years spent on the position)

3.9 Does our training course on writing a CV help people get promoted?

Step 1: Select employees who took CV course

```
1.  SELECT EmpID, Year AS CV_Year
2.  FROM TrainingHistory
3.  WHERE CourseID IN (10, 11, 12);
```

	EmpID	CV_Year
1	001	2017
2	002	2016
3	007	2017
4	009	2018
5	010	2017
6	016	2017
7	023	2017
8	024	2018
9	025	2018
10	030	2017
11	031	2017
12	032	2018
13	033	2018
14	035	2016
15	036	2018
16	037	2017
17	045	2018
18	050	2016
19	052	2017
20	057	2016
21	060	2017
22	064	2018
23	068	2016
24	078	2017
25	079	2018
26	088	2018
27	089	2017
28	092	2017
29	093	2018
30	097	2018
31	099	2016
32	100	2017
33	110	2018
34	114	2016
35	116	2018
36	123	2017
37	127	2018
38	129	2018
39	130	2016
40	131	2017
41	132	2018
42	133	2017

As showed above, there are 43 employees who took and passed the CV course.

Step 2: Use SQL subqueries and inner join to select the promotion year of those employees who took and passed the CV course and got promotion

```
1. SELECT EmpID, YEAR(EndDate) AS Promotion_Year, CV_Year
2. FROM JobHistory
3. INNER JOIN
4. (SELECT EmpID, Year AS CV_Year
5. FROM TrainingHistory
6. WHERE CourseID IN (10, 11, 12)
7. AND PassStatus = 1
8. ) AS CV_Takers
9. USING (EmpID);
```

EmpID	Promotion_Year	CV_Year
009	2015	2018
010	2016	2017
016	2005	2017
016	2012	2017
030	2017	2018
031	2011	2017
032	2013	2018
033	2014	2018
035	2002	2016
035	2010	2016
036	2001	2018
036	2011	2018
037	2003	2017
037	2011	2017
060	2013	2017
064	2011	2018
068	2007	2016
068	2015	2016
093	2012	2018
097	2013	2018
099	2010	2016
099	2015	2016
100	2001	2017
100	2009	2017
123	2016	2017
127	2016	2018
129	2010	2018
130	2005	2016
130	2012	2016
131	2002	2017
131	2010	2017
132	2006	2018
132	2011	2018
133	2004	2017
133	2014	2017

Step 3: Use SQL subqueries and inner join to find the employees who took and passed the CV course and got promotion after then

```
SELECT *
FROM (
SELECT EmpID, YEAR(EndDate) AS Promotion_Year, CV_Year
FROM JobHistory
INNER JOIN
(SELECT EmpID, Year AS CV_Year
FROM TrainingHistory
WHERE CourseID IN (10, 11, 12)
AND PassStatus = 1
) AS CV_Takers
USING (EmpID)
) As CV_Takers_Promoted
WHERE Promotion_Year >= CV_Year;
```

EmpID	Promotion_Year	CV_Year
030	2017	2017

Only 1 out of 43 employees get promotion after taking and passing the CV course. Therefore, we do not have sufficient evidence to argue that the CV course can help the employees get promoted.

3.10 Is our evaluation system working (that is, does it help to develop employees)?

1) Group one: Employees taking courses at least two in 2016

Step 1: Find employees who actively develop themselves in 2016 (store the result as a table "Active_2016")

```
1. SELECT EmpID, COUNT(*) AS Number_of_Courses_Taken
2. FROM TrainingHistory
3. WHERE Year = 2016
4. GROUP BY EmpID
5. HAVING COUNT(*) >= 2
6. ORDER BY COUNT(*) DESC;
```

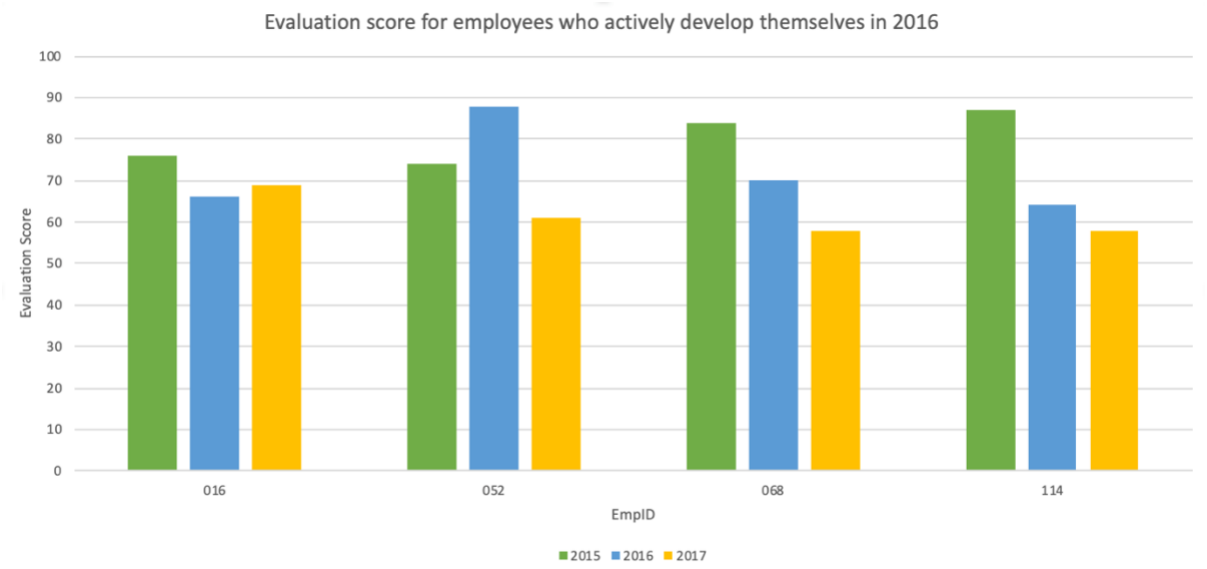
EmpID	Numer_of_Courses_Taken
016	2
052	2
068	2
114	2

Step 2: Check if evaluation score increases in 2016 and 2017 compared to 2015

```
1. SELECT E.EmpID, Year, EvaluationScore
2. FROM Evaluation AS E, Active_2016 AS A
3. WHERE E.EmpID = A.EmpID
4. AND Year IN (2015, 2016, 2017);
```

EmpID	Year	EvaluationScore
016	2015	76
016	2016	66
016	2017	69
052	2015	74
052	2016	88
052	2017	61
068	2015	84
068	2016	70
068	2017	58
114	2015	87
114	2016	64
114	2017	58

After importing the result into Excel, we got the evaluation score diagram:



As showed, after actively taking courses to develop themselves, employees whose EmpID is 016, 068 and 114 receive lower evaluation score in 2016 and 2017 compared to 2015. Therefore, the evaluation system may not positively motivate employees to develop themselves as they take training courses actively in 2016 but keep receiving lower scores.

2) Group two: Employees taking courses at least two in 2017

Step 1: Find employees who actively develop themselves in 2017 (store the result as a table "Active_2017")

```
1. SELECT EmpID, COUNT(*) AS Number_of_Courses_Taken
2. FROM TrainingHistory
3. WHERE Year = 2017
4. GROUP BY EmpID
5. HAVING COUNT(*) >= 2
6. ORDER BY COUNT(*) DESC;
```

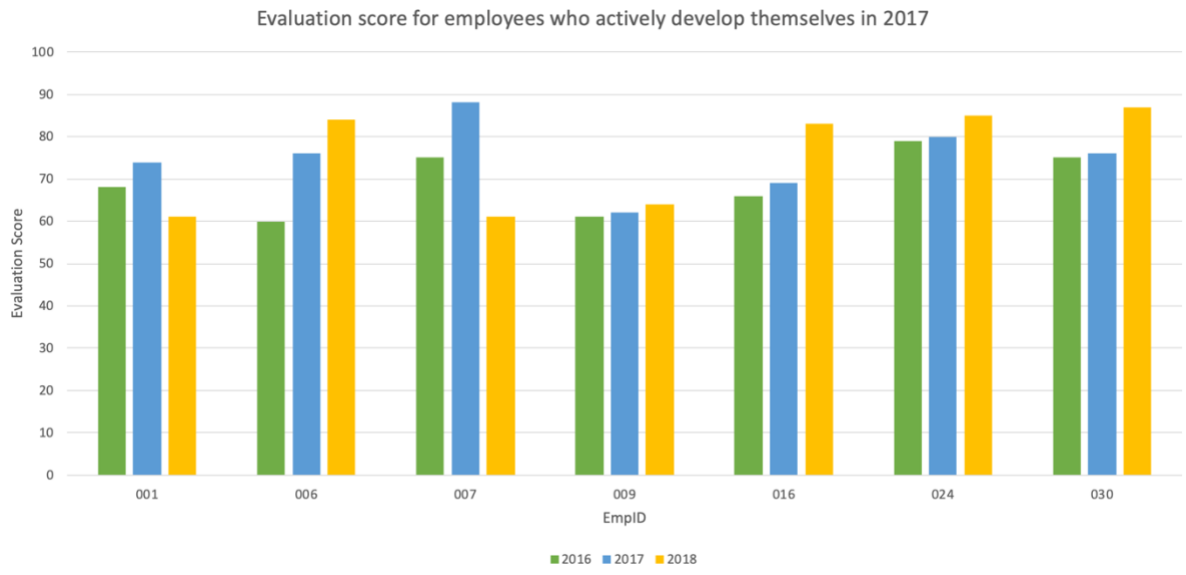
EmpID	Numer_of_Courses_Taken
007	4
001	2
006	2
009	2
016	2
024	2
030	2

Step 2: Check if evaluation score increases in 2016 and 2017 compared to 2015

```
1. SELECT E.EmpID, Year, EvaluationScore
2. FROM Evaluation AS E, Active_2017 AS A
3. WHERE E.EmpID = A.EmpID
4. AND Year IN (2016, 2017, 2018);
```

EmpID	Year	EvaluationScore
001	2016	68
001	2017	74
001	2018	61
006	2016	60
006	2017	76
006	2018	84
007	2016	75
007	2017	88
007	2018	61
009	2016	61
009	2017	62
009	2018	64
016	2016	66
016	2017	69
016	2018	83
024	2016	79
024	2017	80
024	2018	85
030	2016	75
030	2017	76
030	2018	87

After importing the result into Excel, we got the evaluation score diagram:



As showed, all employees who actively take courses in 2017 receive higher evaluation marks in 2017 compared to the previous year. In addition, five out seven employees keep receiving marks in 2018 after actively developing themselves in 2018. Therefore, the evaluation system helps employees develop themselves in 2017.

3) Group three: Employees taking courses at least two in 2018

Step 1: Find employees who actively develop themselves in 2018 (store the result as a table "Active_2018")

```
7.  SELECT EmpID, COUNT(*) AS Number_of_Courses_Taken
8.  FROM TrainingHistory
9.  WHERE Year = 2017
10. GROUP BY EmpID
11. HAVING COUNT(*) >= 2
12. ORDER BY COUNT(*) DESC;
```

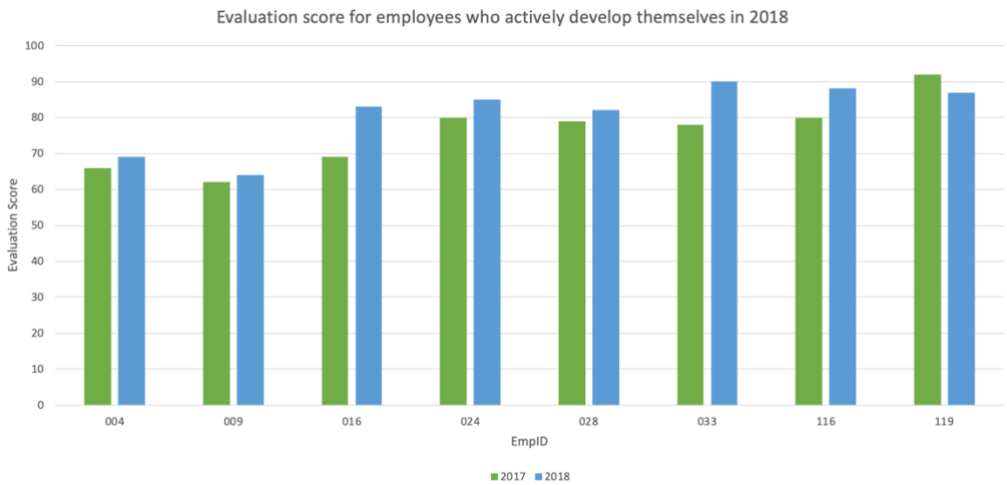
EmpID	Number_of_Courses_Taken
119	3
116	2
004	2
009	2
016	2
024	2
028	2
033	2

Step 2: Check if evaluation score increases in 2016 and 2017 compared to 2015

```
5.  SELECT E.EmpID, Year, EvaluationScore
6.  FROM Evaluation AS E, Active_2018 AS A
7.  WHERE E.EmpID = A.EmpID
8.  AND Year IN (2017, 2018);
```

EmpID	Year	EvaluationScore
004	2017	66
004	2018	69
009	2017	62
009	2018	64
016	2017	69
016	2018	83
024	2017	80
024	2018	85
028	2017	79
028	2018	82
033	2017	78
033	2018	90
116	2017	80
116	2018	88
119	2017	92
119	2018	87

After importing the result into Excel, we got the evaluation score diagram:



As showed, seven out of eight employees who actively develop themselves in 2018 receive a higher evaluation mark in 2018. Therefore, the evaluation system helps employees develop themselves in 2018.

In conclusion, the evaluation system helps employees generally, and it gradually perform better in motivating employees to actively take training courses.

PART FOUR: WHY DO OUR EMPLOYEES LEAVE US?

The problem would be analysed using R. Load libraries and the data first:

```
library(plyr)
library(dplyr)            #filter()
library(corrplot)         #corrplot()
library(ggplot2)          #ggplot()
library(gridExtra)        #grid.arrange()
library(Rmisc)            #multiplot()
library(e1071)            #naiveBayes()
library(rpart)            #raprt()
library(pROC)             #roc()
library(rpart.plot)       #raprt.plot()
library(randomForest)     #randomForest()
library(caret)            #confusionMatrix()
library(DT)               #datatable()
hr <- read.csv("../input/HRSurveyData.csv", stringsAsFactors = FALSE)
```

4.1 Understand Data

```
str(hr)
```

Result:

```
'data.frame': 14999 obs. of  10 variables:
 $ satisfaction_level   : num  0.38 0.8 0.11 0.72 0.37 0.41 0.1 0.92 0.89 0.42 ...
 $ last_evaluation      : num  0.53 0.86 0.88 0.87 0.52 0.5 0.77 0.85 1 0.53 ...
 $ number_project       : int  2 5 7 5 2 2 6 5 5 2 ...
 $ average_montly_hours : int  157 262 272 223 159 153 247 259 224 142 ...
 $ time_spend_company   : int  3 6 4 5 3 3 4 5 5 3 ...
 $ Work_accident        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ left                 : int  1 1 1 1 1 1 1 1 1 1 ...
 $ promotion_last_5years: int  0 0 0 0 0 0 0 0 0 0 ...
 $ sales                : Factor w/ 10 levels "accounting","hr",..: 8 8 8 8 8 8 8 8 8 8 ...
 $ salary               : Factor w/ 3 levels "high","low","medium": 2 3 3 2 2 2 2 2 2 2 ...
```

The data set contains ten attributes and has 14999 records. These ten attributes will be used to analyse employee attrition and we would build models to predict the turnover in part five of the report.

4.2 Data Pre-processing

Rename variables

There is a typo 'montly' in the dataset and we need to correct it and then simplify variables names.

```
hr <- rename(hr,
   satisfaction = satisfaction_level, evaluation = last_evaluation,
   project = number_project, monthlyhour = average_montly_hours,
   serviceyear = time_spend_company, accident = Work_accident,
   promotion = promotion_last_5years, dept = sales)
```

Transfer the variable "left" into a factor variable

```
hr$left <- factor(hr$left, levels = c("0","1"))
```

Add a new variable salarynum

```
hr$salarynum[hr$salary == "low"] <- 1
hr$salarynum[hr$salary == "medium"] <- 2
hr$salarynum[hr$salary == "high"] <- 3


summary(hr)
```

Result:

```
 satisfaction_level last_evaluation  number_project  average_montly_hours time_spend_company
 Min.   :0.0900     Min.   :0.3600   Min.   :2.000   Min.   : 96.0        Min.   : 2.000
 1st Qu.:0.4400     1st Qu.:0.5600   1st Qu.:3.000   1st Qu.:156.0        1st Qu.: 3.000
 Median :0.6400     Median :0.7200   Median :4.000   Median :200.0        Median : 3.000
 Mean   :0.6128     Mean   :0.7161   Mean   :3.803   Mean   :201.1        Mean   : 3.498
 3rd Qu.:0.8200     3rd Qu.:0.8700   3rd Qu.:5.000   3rd Qu.:245.0        3rd Qu.: 4.000
 Max.   :1.0000     Max.   :1.0000   Max.   :7.000   Max.   :310.0        Max.   :10.000

 Work_accident         left        promotion_last_5years         sales         salary
 Min.   :0.0000   Min.   :0.0000   Min.   :0.00000       sales      :4140   high  :1237
 1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.00000       technical  :2720   low   :7316
 Median :0.0000   Median :0.0000   Median :0.00000       support    :2229   medium:6446
 Mean   :0.1446   Mean   :0.2381   Mean   :0.02127       IT         :1227
 3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.00000       product_mng: 902
 Max.   :1.0000   Max.   :1.0000   Max.   :1.00000       marketing  : 858
 salarynum
 Min.   :1.000
 1st Qu.:1.000
 Median :2.000
 Mean   :1.595
 3rd Qu.:2.000
 Max.   :3.000
```

4.3 Variable Description

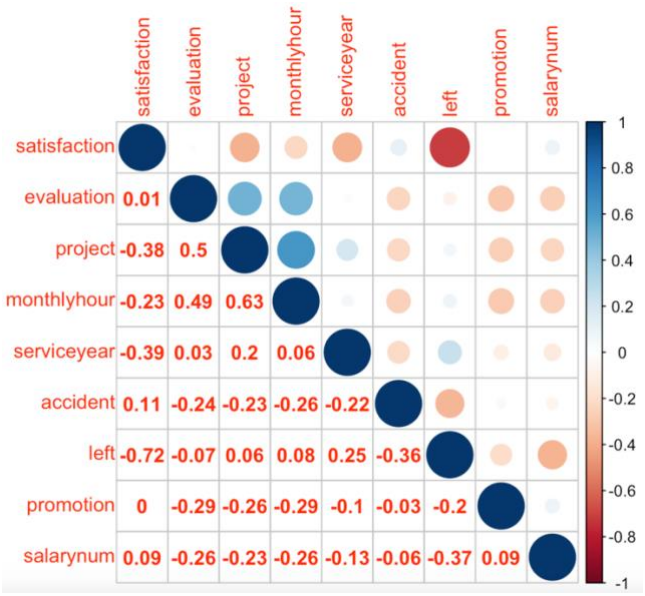
Based on the elementary analysis, we produce a table "Variable Description" as a data dictionary to summary the variables information:

Variable Type	Variable Name	Abbreviation	Definition	Range	Note
Dependent	left	left	Whether the employee has left	0 for has not left, 1 for left	23.81% employees has left
Independent	satisfaction_level	satisfaction	Employee's satisfaction level	0~1	0.6 on average
	last_evaluation	evaluation	Employee's last evaluation	0~1	0.7 on average
	number_project	project	Number of projects each employee have worked on	2~7	4 on average
	average_monthly_hours	monthlyhour	Average hours working each month	96~310	
	time_spend_company	serviceyear	Number of years each employee spent at the company	2~10	
	Work_accicdent	accident	Whether the employee have had an accident at work	0 for has not had an accident, 1 for has had an accident	14.46% has an accident on average
	promotion_last_5years	promotion	Whether the employee have had a promotion in the last 5 years	0 for not had a promotion, 1 for had a promotion	21.27% has promotion on average
	sales	dept	Employee's department	sales, hr, accounting etc.	
	salary	salary	Employee's salary	low, medium, high	48.78% for low, 8.25% for high
	salarynum	salarynum	Employee's salary	0, 1, 2	48.78% for low, 8.25% for high

4.4 Data Analysis and Visualisation: Correlation Matrix

For any two variables X and Y, the correlation between them represents the degree of linearly relation of X and Y (Lu, 2019). Correlation matrix show the relationship between multiple variables intuitively in a table (Lu, 2019).

```
1. hr_cor <- hr %>%
2. select(satisfaction:promotion, salarynum)
3. corrmatrix <- cor(hr_cor)
4. corrplot.mixed(corrmatrix)
5. corrplot(cor(corrmatrix),
   type="upper",method = "circle",tl.pos = "tl",tl.offset = 0.05)
6. corrplot(cor(corrmatrix),
   add=T,type="lower",method = "number",col="red",diag=F, tl.pos ="n",cl.pos ="n")
```



The size of circle represents the size of the correlation and the colour represents two variables are positively or negatively correlated. As the correlation matrix showed, the factors related to employee attrition are satisfaction level (-0.72), salary (-0.37), work accident (-0.36), promotion (-0.2), last evaluation (-0.07), number of projects involved (0.06), monthly working hours (0.08), serve year (0.25). In general, employees choose to leave the company because low satisfaction level, low salary level, no promotion opportunity, long serve year and long monthly working hours.

4.5 Data Analysis and Visualisation: Exploratory Data Analysis

Transfer variable data type into factor data type

```
1. hr$accident <- as.factor(hr$accident)
2. hr$left <- as.factor(hr$left)
3. hr$promotion <- as.factor(hr$promotion)
4. hr$dept <- as.factor(hr$dept)
5. hr$salary <- as.factor(hr$salary)
```

Analysing categorical variables

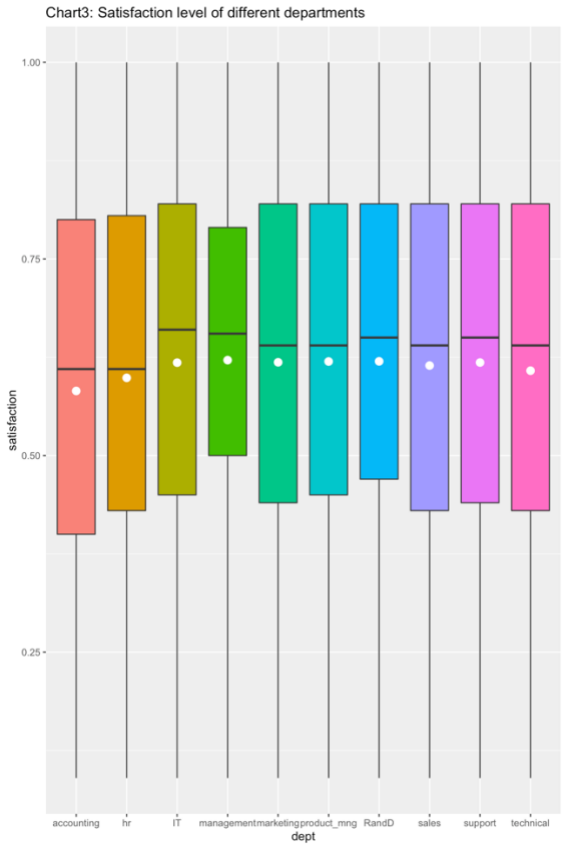
```
g1 <- ggplot(group_by(hr, dept), aes(x = dept, fill = dept)) + geom_bar(width = 1) +
      coord_polar(theta = "x") + ggtitle("Chart1: Number of employees in different departments")
g2 <- ggplot(group_by(hr, dept), aes(x = dept, fill = salary)) + geom_bar(width = 1) +
      coord_polar(theta = "x") + ggtitle("Chart2: Salary status in different departments")
multiplot(g1, g2, cols = 2)
```

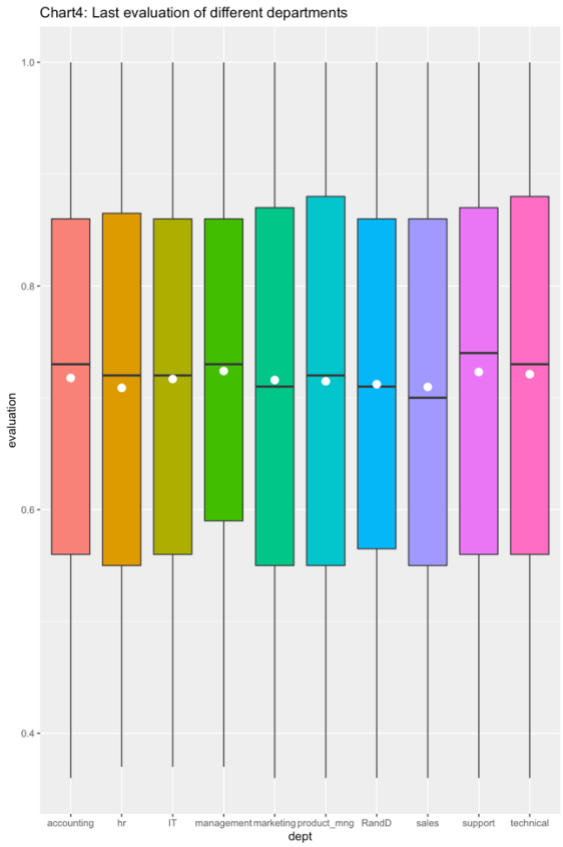
1) The employees in sales, support and technical departments are dominant in terms of population in this company.

2) Except management, more employees working in other departments get medium- and high-level salary as the population in department increases.

Analysing categorical and numerical variables

```
g3 <- ggplot(hr, aes(x = dept, y = satisfaction, fill = dept)) + geom_boxplot() +
      ggtitle("Chart3: Satisfaction level of different departments") +
      stat_summary(fun.y = mean,size = 3, color = 'white', geom = "point") + theme(legend.position = "none")
g4 <- ggplot(hr, aes(x = dept, y = evaluation, fill = dept)) + geom_boxplot() +
      ggtitle("Chart4: Last evaluation of different departments") +
      stat_summary(fun.y = mean,size = 3, color = 'white', geom = "point") + theme(legend.position = "none")
grid.arrange(g3, g4, ncol = 2)
```



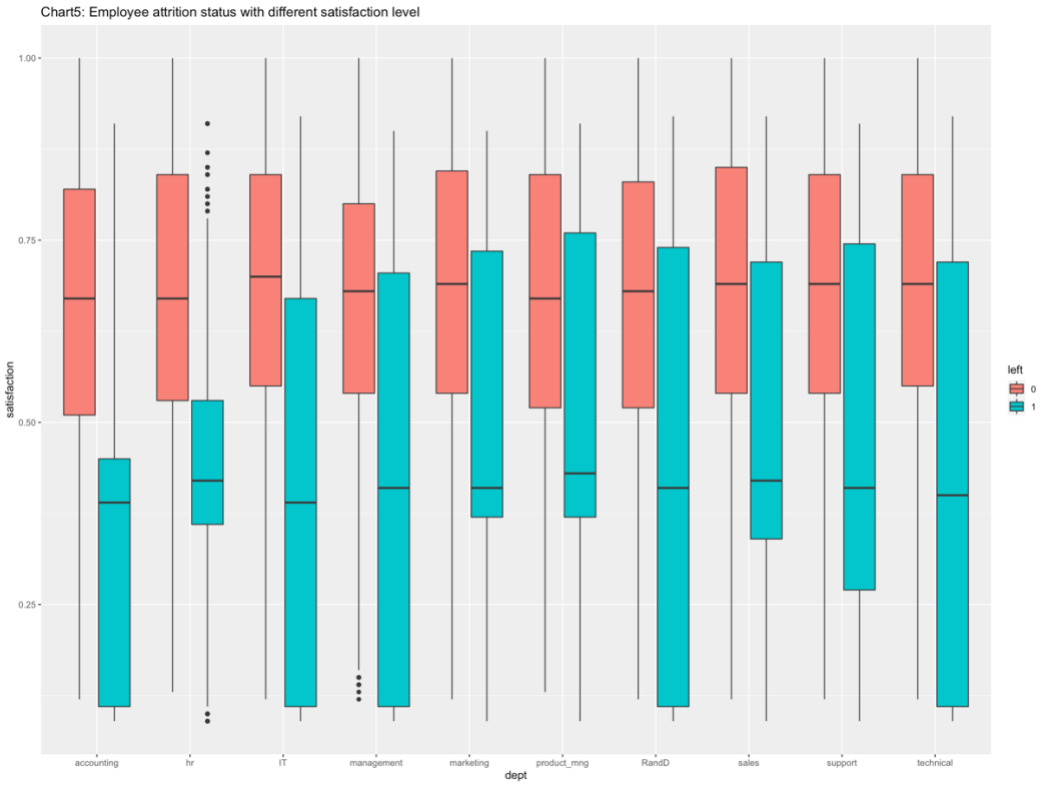


1) Employees in accounting department get lower satisfaction and satisfaction level is biggest employee turnover factor analysed in correlation matrix. This indicates that low satisfaction maybe the major reason for accounting employee attrition.

2) For last evaluation score, it is pretty similar among different departments.

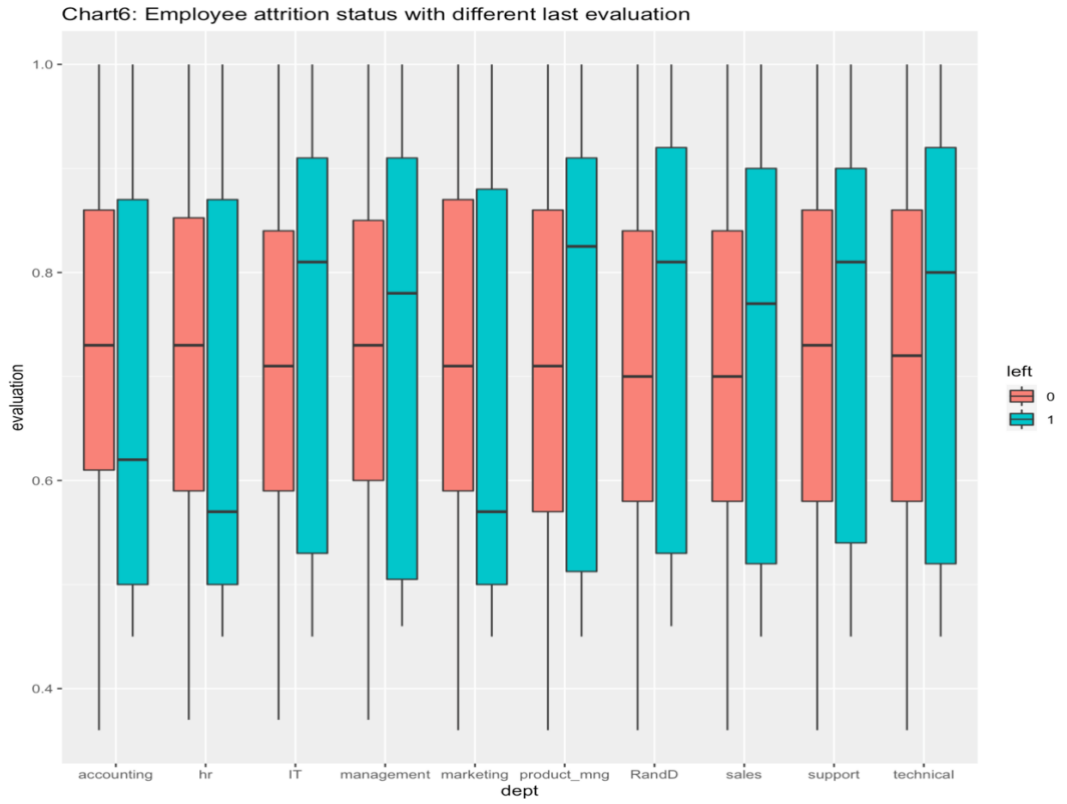
Analysing possible reasons of employee attrition

```
1. ggplot(hr, aes(x = dept, y = satisfaction, fill = left)) + geom_boxplot() +
2. ggtitle("Chart5: Employee attrition status with different satisfaction level")
```



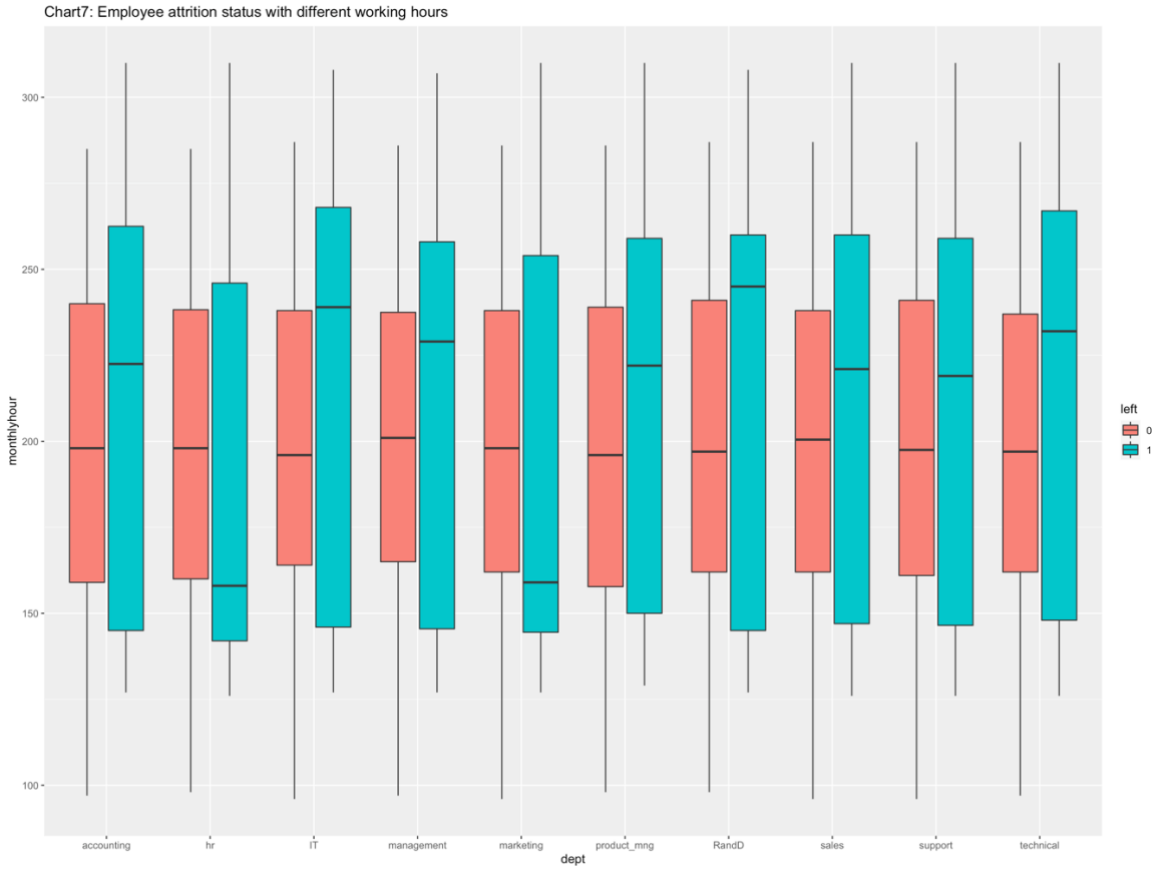
1) People who leave the company normally get low satisfaction level for all departments.

```
3. ggplot(hr, aes(x = dept, y = evaluation, fill = left)) + geom_boxplot() +
4. ggtitle("Chart6: Employee attrition status with different last evaluation")
```



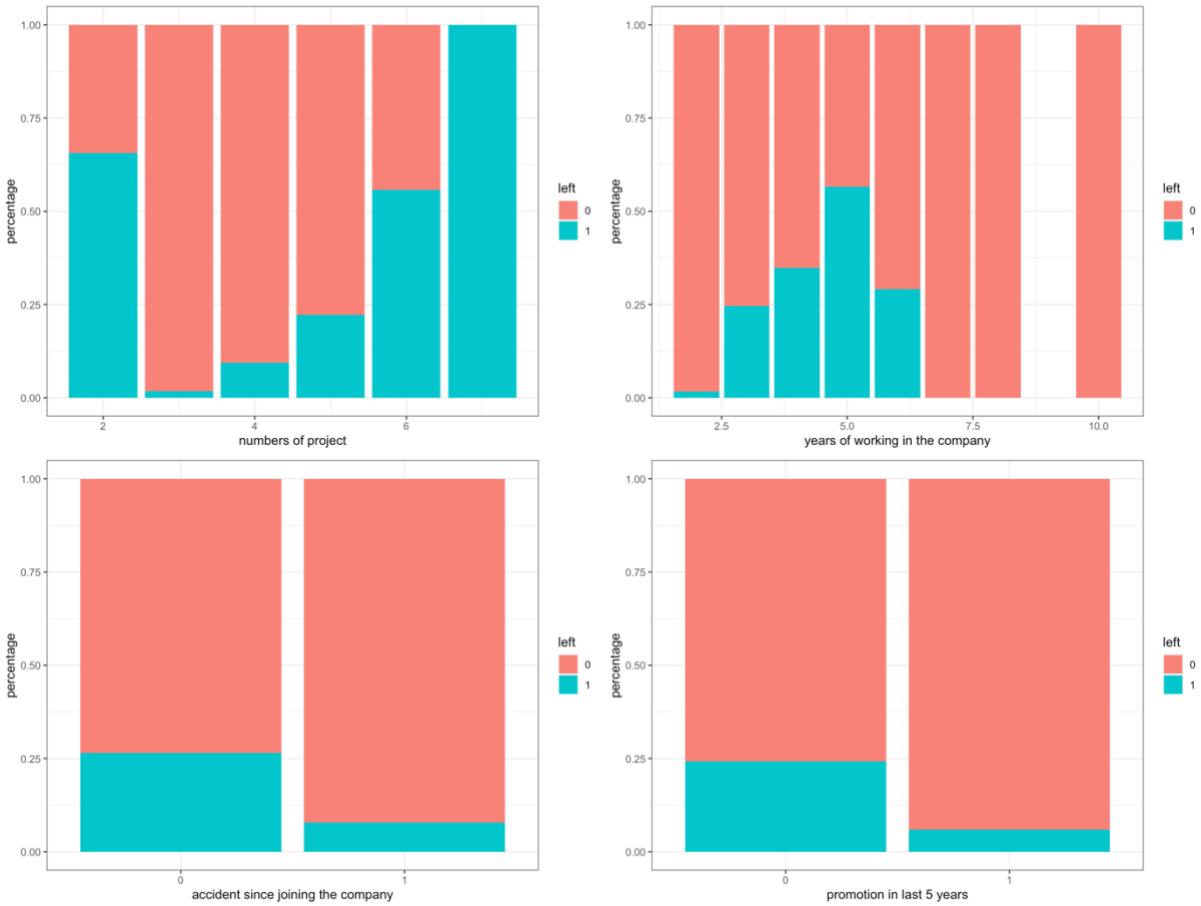
1) Employees in accounting, hr and marketing departments get low last evaluation score and they finally leave.

```
5.  ggplot(hr, aes(x = dept, y = monthlyhour, fill = left)) + geom_boxplot() +
6.  ggtitle("Chart7: Employee attrition status with different working hours")
```



1) For people who still stay in the company, they work in a stable pace, with similar monthly working hours. However, for employees in hr and marketing departments, those who do not work hard finally leave.

```r
b1 <- ggplot(hr, aes(x = project, fill = left)) +
      geom_bar(position = 'fill') +
      theme_bw() + labs(x = 'numbers of project', y = 'percentage')
b2 <- ggplot(hr, aes(x = serviceyear, fill = left)) +
      geom_bar(position = 'fill') +
      theme_bw() + labs(x = 'years of working in the company', y = 'percentage'
)
b3 <- ggplot(hr, aes(x = accident, fill = left)) +
      geom_bar(position = 'fill') +
      theme_bw() + labs(x = 'accident since joining the company', y = 'percenta
ge')
b4 <- ggplot(hr, aes(x = promotion, fill = left)) +
      geom_bar(position = 'fill') +
      theme_bw() + labs(x = 'promotion in last 5 years', y = 'percentage')
grid.arrange(b1, b2, b3, b4, ncol = 2, nrow = 2)
```

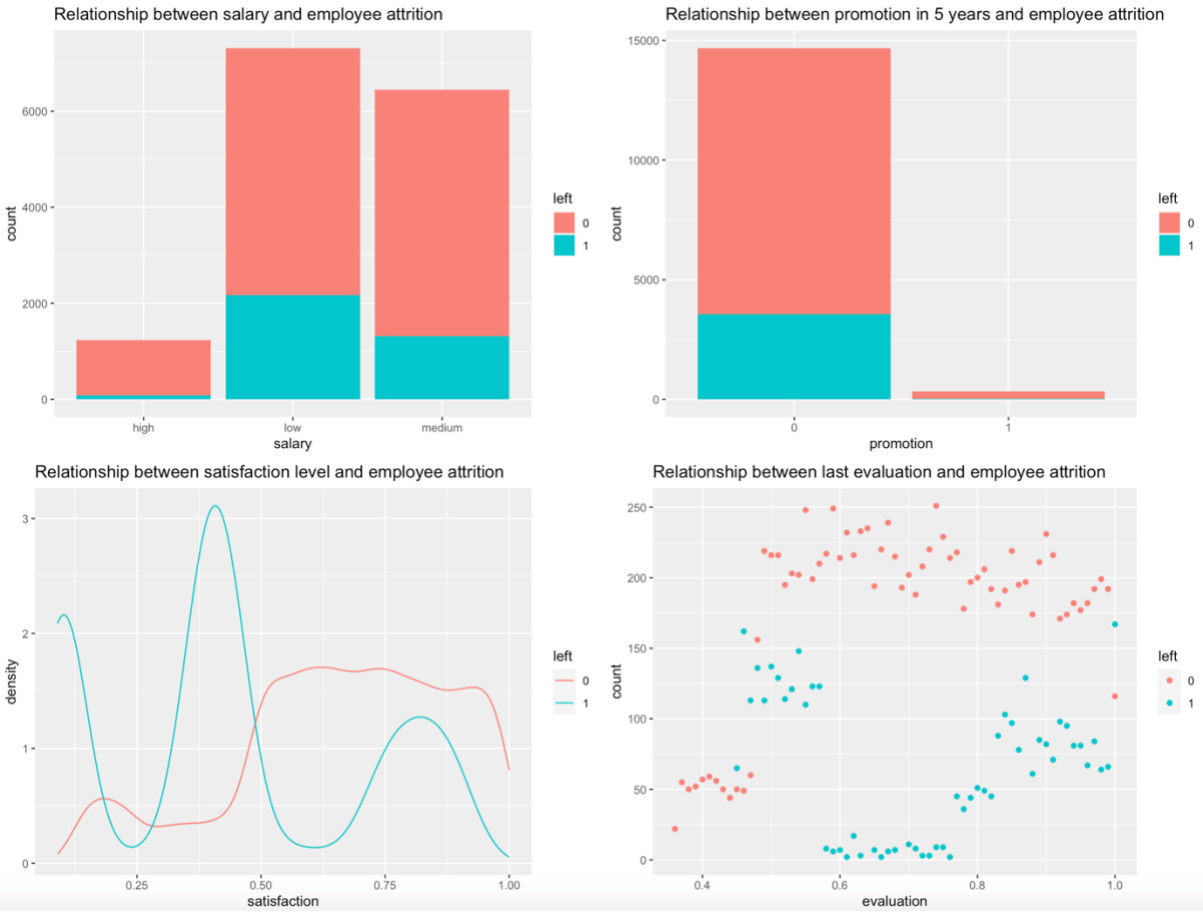


1) After involving three projects, more projects finished, more likely to leave.

2) People who work for three to six years in the company have highest turnover rate.

3) Employees who do not have accident and do not get promotion in last 5 years have higher turnover rate.

```
1. b6 <- ggplot(hr, aes(x = salary, fill = left)) + geom_histogram(stat = "count")
       + ggtitle("Relationship between salary and employee attrition")
2. b7 <- ggplot(hr, aes(x = promotion, fill = left)) +
         geom_histogram(stat = "count") +
   ggtitle("Relationship between promotion in 5 years and employee attrition")
3. b8 <- ggplot(hr, aes(x = satisfaction, color = left)) +
         geom_line(stat = "density") +
   ggtitle("Relationship between satisfaction level and employee attrition")
4. b9 <- ggplot(hr, aes(x = evaluation, color = left)) +
         geom_point(stat = "count") +
   ggtitle("Relationship between last evaluation and employee attrition")
5. grid.arrange(b6, b7, b8, b9, ncol = 2, nrow = 2)
```



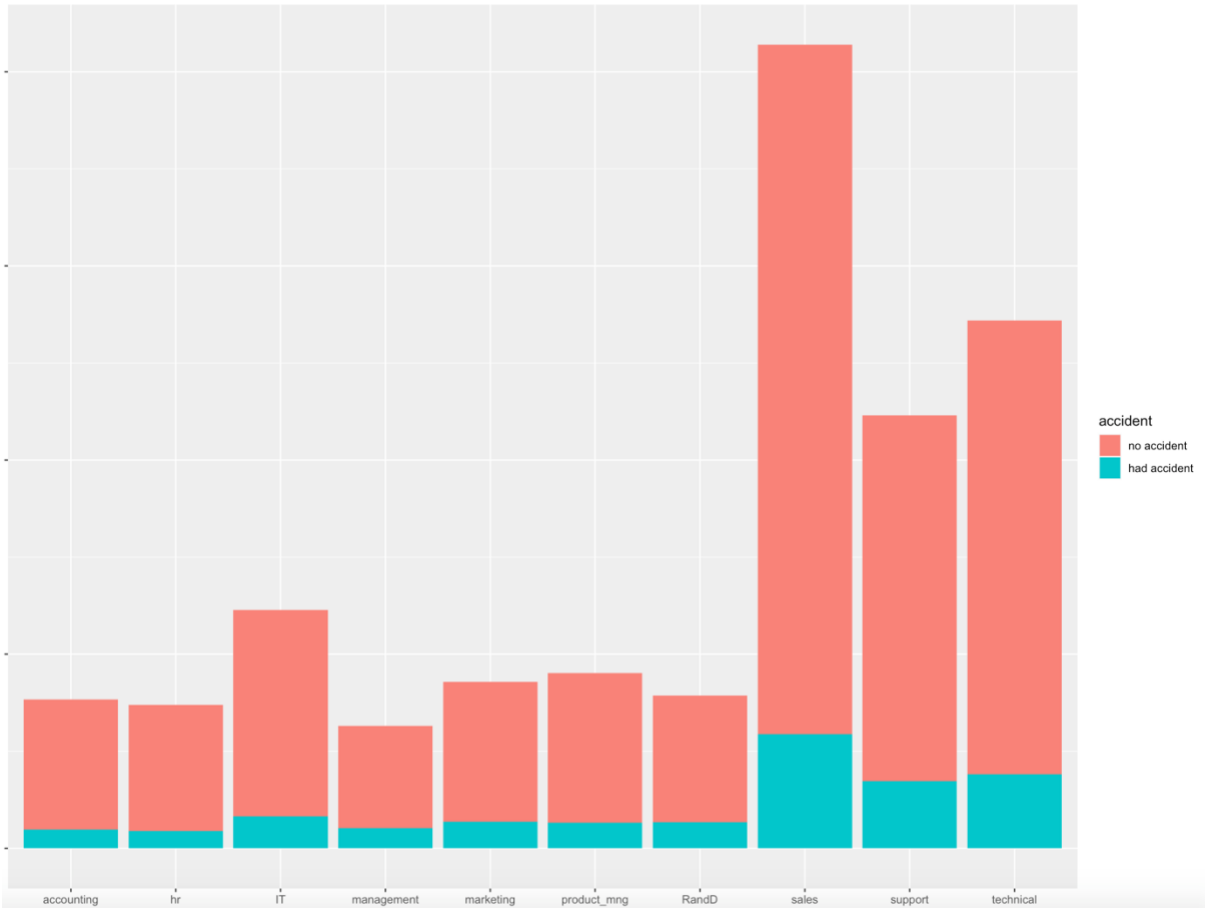
1) Employees who get high salary rarely leave.

2) If employees get promotion in the recent 5 years, most of them will not leave.

3) Retired employees rate satisfaction score in different regions, so we cannot say that employees who get low satisfaction level will have high probability of leaving.

4) A lot of retired employees will get high last evaluation score, so we cannot argue that those who get high evaluation level will probably not leave.

```
hr %>%
group_by(dept) %>%
ggplot(aes(y = dept,fill = accident)) + geom_bar() + coord_flip() +
theme(axis.text.y = element_blank(), axis.title.y = element_blank(),
axis.title.x = element_blank()) +
scale_fill_discrete(labels = c("no accident", "had accident"))
```



1) The number of people in each department who had accident during work is positive correlated with the population of the department. Therefore, accident is not a major reason for leaving the company.

4.6 Data Analysis Summary

The major reasons of employee attrition are listed below:

1) Low salary: High salary's employees rarely leave.
2) Overload work time: Overload monthly working hours, but for employees in hr and marketing departments, they will leave because of too little working hours.
3) Strong working ability: After participating in 2 project, ore projects participated afterward, more likely to leave. About half of those whose last evaluation score is high leave. Those who stay in the company for three to six years are likely to leave.
4) No promotion in last 5 years: If getting promotion, very few employees leave. If not, over 2/7 $\approx$ 28.6% of employees leave.

PART FIVE: MODELING TO PREDICT EMPLOYEE ATTRITION

Predicting employee turnover is a classification problem, with two classes of leave or not leave. Three classification models would be built and evaluated under AUC and confusion matrix. Then, we would choose the best model to build an interactive table for managers to check which employee is predicted to leave.

5.1 Naive Bayes Model

Naive Bayes is based on Bayes' theorem, under the assumption that there is an independent relationship between predictor variables and other variables (Bhartiya et al., 2019). The model is suitable for large datasets and it utilise the probabilities of attributes to make predictions (Bhartiya et al., 2019).

Splitting the data into training and test set

```
1. set.seed(1234)
2. n <- nrow(hr)
3. rnd <- sample(n, n*.70)
4. train <- hr[rnd,]
5. test <- hr[-rnd,]
```

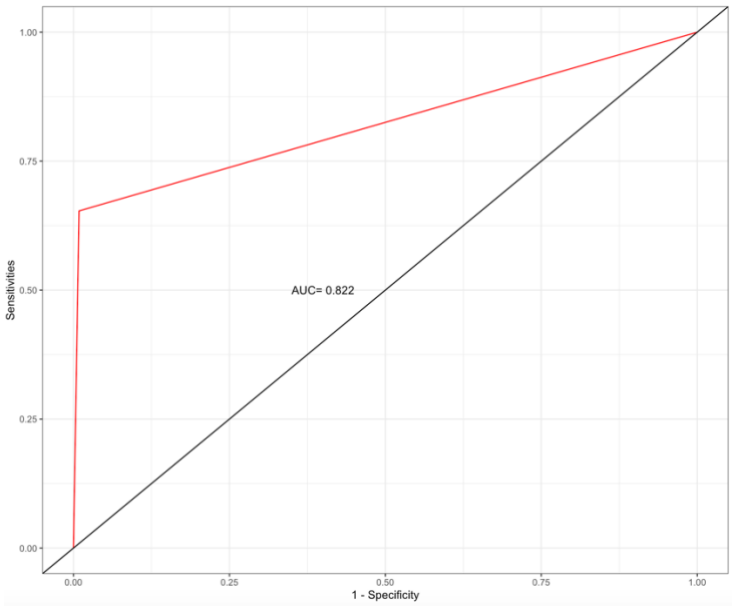
Build Naive Bayes Model

```
1. train_control <- trainControl(method = 'cv', number = 5)
2. nbmodel <- train(left ~ ., data = train,
   trControl = train_control, method = 'nb')
3. pred_nb <- predict(nbmodel, test[-7])
```

Build ROC Curve and check the model's AUC

Receiver operating characteristics (ROC) curve is a graph to general evaluate the prediction accuracy of machine learning models (Fawcett, 2006). The area under ROC curve is named as Area under the ROC curve (AUC) and the higher AUC means the higher accuracy of prediction (Fawcett, 2006). AUC will be used as one criterion to evaluate the general performance of our models.

```
pred_nb <- as.numeric(as.character(pred_nb))
roc_nb <- roc(test$left, pred_nb)
Specificity <- roc_nb$specificities
Sensitivity <- roc_nb$sensitivities
ggplot(data = NULL, aes(x = 1- Specificity, y = Sensitivity)) +
          geom_line(colour = 'red') + geom_abline() +
          annotate('text', x = 0.4, y = 0.5, label = paste('AUC=',
          round(roc_nb$auc, 3))) + theme_bw() +
          labs(x = '1 - Specificity', y = 'Sensitivities')
```



The Naive Bayes Model has AUC of 0.822, which is higher than 0.75 and this indicates the model is a good model to do prediction.

Use confusion matrix to check the model's accuracy

Confusion matrix is our second criterion to evaluate classification models. According to Rohit Hebbar et al. (2018), confusion matrix is commonly used to find the best performing machine learning model and it is formatted like the graph showed below. The corresponding illustration of terminologies is also listed. In our case, TP and TN represent the model correctly predict situations that employees who would not leave and would leave the company. Therefore, TP and TN are important indicators of our model's detailed accuracy in predicting the two situations.

Actual/Predicted	0	1
0	TN	FP
1	FN	TP

	Meaning
True Positive (TP)	the observation is predicted to be positive when the actual observation is positive
False Negative (FN)	the observation is predicted to be negative when the actual observation is positive
False Positive (FP)	the observation is predicted to be positive when the actual observation is negative
True Negative (TN)	the observation is predicted to be negative when the actual observation is negative

```
1.  prop.table(table(pred_nb, test$left, dnn = c("Actual", "Predicted")), 1)
```

Result:

```
1.        Predicted
2.  Actual          0          1
3.       0 0.75505668 0.08246277
4.       1 0.00689042 0.15559013
```

With the accuracy rate of near 75.5%, the model is a good one when predict that employees do not leave the company. However, it does not perform well when predict employee turnover, as it only gets around 15.6% accuracy rate. This indicates that we should try other models.

5.2 Decision Tree Model

Decision tree is used to find the correlations between attributes in an entire dataset and the corresponding graph can be produced as a decision support tool (Chesney, 2009). A Decision Tree Model would be built and accessed using AUC and confusion matrix as before.

Build the Decision Tree Model

```
dtree <- rpart(left ~., data = train)
preds <- predict(dtree, test, type = "class")
```

Build ROC Curve and check the model's AUC

```
rocv <- roc(as.numeric(test$left), as.numeric(preds))
rocv$auc
```

Result:

```
Area under the curve: 0.9524
```

The AUC is above 90%, which indicate the Decision Tree Model is much better than the Naive Bayes Model.

Use confusion matrix to check the model's accuracy

```
prop.table(table(test$left, preds, dnn = c("Actual", "Predicted")), 1)
```

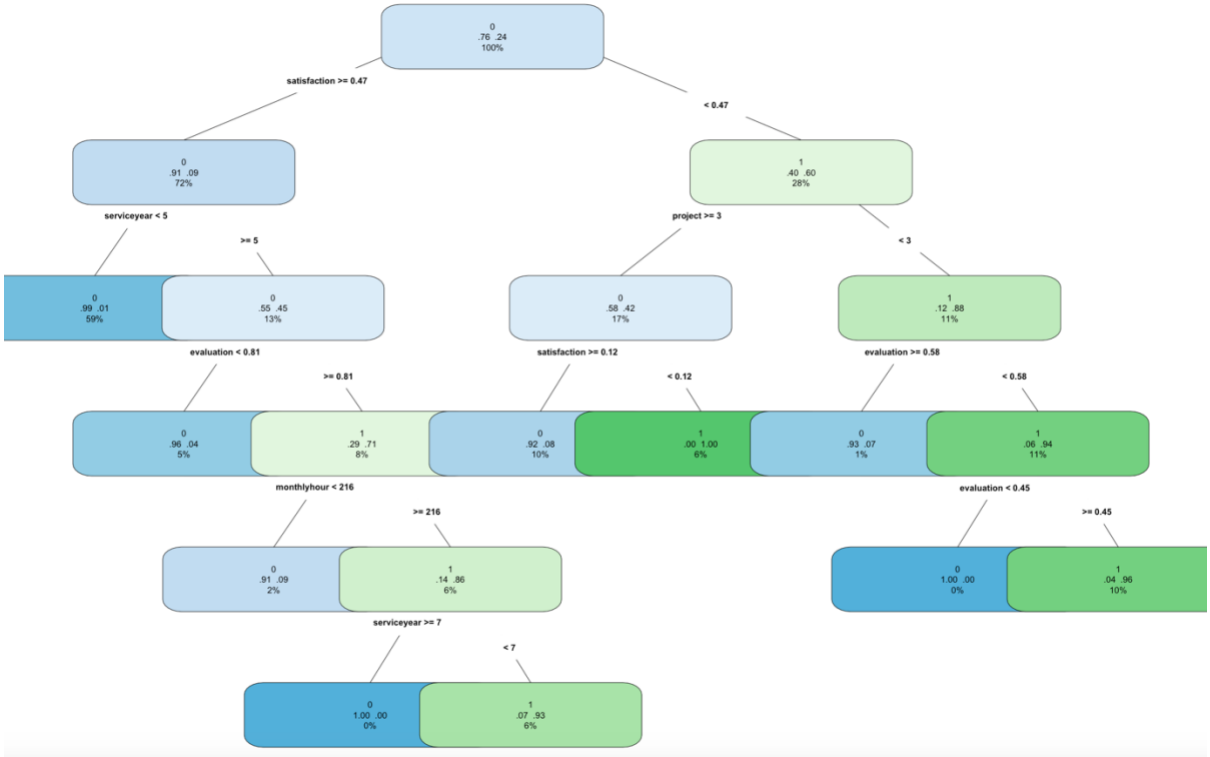
Result:

```
        Predicted
Actual          0          1
     0 0.98861646 0.01138354
     1 0.08379888 0.91620112
```

Again, the model has high accuracy of predicting both employees who leave and do not leave the company. Therefore, we can conclude that the Decision Tree Model is better than Naive Bayes Model in this case.

Draw the decision tree diagram

```
1. rpart.plot(dtree,
2.              type = 4,
3.              extra = 104,
4.             tweak = 0.8,
5.              fallen.leaves = F,
6.              cex = 0.7)
```



As the decision tree showed above, satisfaction is the most important factor of employee attrition. Number of projects involved, serve year, monthly working hours, last evaluation are also important factors. This Decision Tree can help managers predict an employee will leave or not intuitively.

5.3 Random Forest Model

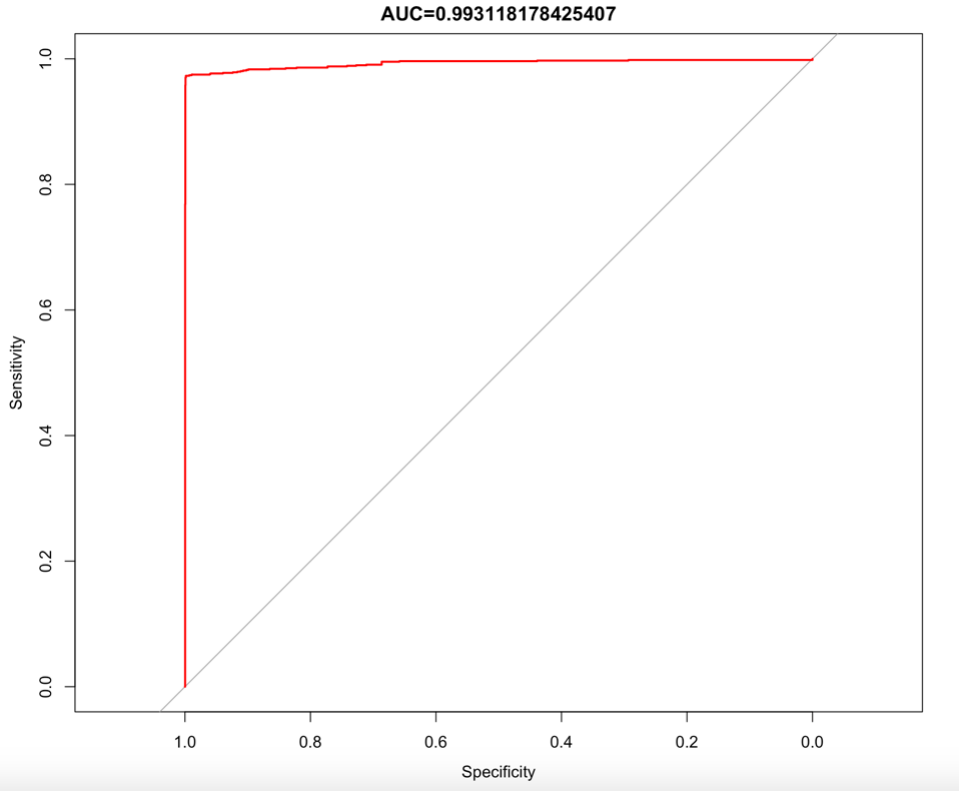
Random Forest builds many decision trees and then combine them to perform a better prediction (Bhartiya et al., 2019). A Random Forest Model would be established and evaluated using the same criteria as before.

Build Random Forest Model

```
1. index <- sample(2, nrow(hr), replace = T, prob = c(0.7,0.3))
2. train <- hr[index == 1,]; test <- hr[index == 2,]
3. rfmodel <- randomForest(left~., data = train)
4. predict.hr <- predict(rfmodel, test)
```

Build ROC Curve and check the model's AUC

```
1. prob.hr <- predict(rfmodel, test, type = "prob")
2. roc.hr <- roc(test$left, prob.hr[,2], levels = levels(test$left))
3. plot(roc.hr,type="S", col ="red", main = paste("AUC=", roc.hr$auc, sep = ""))
```



The AUC is extremely high, so among the tree models above, this company should choose the Random Forest Model to predict employee attrition.

Use confusion matrix to check the model's accuracy

```
confusionMatrix(test$left, predict.hr)
```

Result:

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 3362    3
         1   32 1039

               Accuracy : 0.9921
                 95% CI : (0.989, 0.9945)
    No Information Rate : 0.7651
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9783

 Mcnemar's Test P-Value : 2.214e-06

            Sensitivity : 0.9906
            Specificity : 0.9971
         Pos Pred Value : 0.9991
         Neg Pred Value : 0.9701
             Prevalence : 0.7651
         Detection Rate : 0.7579
   Detection Prevalence : 0.7586
      Balanced Accuracy : 0.9938

       'Positive' Class : 0
```

Again, Random Forest Model is the best one, having extremely high accuracy in predicting employees who would leave and not leave.

5.4 Interactive Table of Predicting Employee Attrition

```
1. predict.hr <- as.numeric(as.character(predict.hr))
2. pred_end <- predict(rfmodel, test[-7], type = 'prob')
3. data_end <- cbind(round(pred_end, 3), predict.hr)
4. datatable(data_end)
```

Show 25 entries Search:

	0	1	predict.hr
2	0.032	0.968	1
6	0	1	1
7	0	1	1
10	0	1	1
11	0	1	1
12	0	1	1
14	0	1	1
16	0	1	1
18	0.002	0.998	1
19	0.066	0.934	1
23	0	1	1
25	0	1	1
31	0.01	0.99	1
32	0	1	1
34	0	1	1
36	0.002	0.998	1
38	0.002	0.998	1
43	0	1	1

Showing 1 to 25 of 4,436 entries

Previous 1 2 3 4 5 … 178 Next

Based on the Random Forest Model which has 99.3% accuracy rate, we build an interactive table. In this table, 1 for predicting employee turnover and 0 for predicting employees will not leave, and 'predict.hr' represent the prediction result. For example, employee whose id is 2 has 96.8% for leaving and this employee actually leave. This table would be useful for managers to check which employee would leave intuitively and the corresponding possibility. After knowing which employees would leave, and the major reasons analysed in the previous section, managers may take actions to prevent employees from leaving efficiently.

REFERENCES

Bhartiya N., Jannu S., Shukla P. and Chapaneri R. (2019), 'Employee Attrition Prediction Using Classification Models'*, 2019 IEEE 5th International Conference for Convergence in Technology (I2CT),* Bombay, India, pp.1-6.

Chesney, T. (2009) *Searching For Patterns: How We Can Know without Asking*. Nottingham: Nottingham University Press.

Fawcett, T. (2006), 'An introduction to ROC analysis', *Pattern Recognition Letters,* 27(8), pp.861-874.

Lu, B. (2019), *Demystifying The Correlation Matrix,* Data Driven Investor. Available at: https://www.datadriveninvestor.com/2019/04/24/demystifying-the-correlation-matrix/ (Accessed: 01 May 2020)

Rohit Hebbar A., Patil S. H., Rajeshwari S. B. and Saqquaf, S. S. M. (2018), 'Comparison of Machine Learning Techniques to Predict the Attrition Rate of the Employees', *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT),* Bangalore, India, pp. 934-938