

# Database Design for School Management System

Final Project for Advanced Database Management System Group 3

ISM6218.003F21.96801

Advanced Database Management

## Team Members

Gurram Rupa – U55600349

Abhishek Malga - U82138357

Lakshmi Sai Prasanna Boddu – U33015229

Siddhartha Ananthaneni – U72616874

Chaitanya Sai Kopparthi – U93449053



## **Table of Contents**

<b>Serial No.</b>	<b>Title</b>	<b>Pag e No .</b>
<b>1.</b>	<b>Purpose</b>	<b>4</b>
<b>2.</b>	<b>Narrative</b>	<b>4</b>
<b>3.</b>	<b>Entities with separate records</b>	<b>5</b>
<b>4.</b>	<b>Entities Attributes</b>	<b>6</b>
<b>5.</b>	<b>Entity Relationship Diagram (ERD)</b>	<b>7</b>
<b>6.</b>	<b>Table Views</b>	<b>8</b>
<b>7.</b>	<b>Data Integrity</b>	<b>11</b>
<b>8.</b>	<b>SQL Queries</b>	<b>14</b>
<b>9.</b>	<b>Performance Tuning</b>	<b>18</b>
<b>10.</b>	<b>Data Visualization</b>	<b>20</b>

<b>Topic Area</b>	<b>Description</b>	<b>Group Member</b>	<b>Weight</b>
<b>Databas e Design</b>	<b>This part should include a logical database design (for the relational model), using normalization to control redundancy and integrity constraints for data quality.</b>	<b>Lakshmi Chaitanya</b>	<b>25%</b>
<b>Query Writing</b>	<b>This part is another chance to write SQL queries, explore transactions.</b>	<b>Abhishek Rupa</b>	<b>25%</b>
<b>Performan ce Tuning</b>	<b>In this section, you can capitalize and extend your prior experiments with indexing, optimizer modes, partitioning, parallel execution.</b>	<b>Lakshmi Siddhartha</b>	<b>25%</b>
<b>DBA Scripts</b>	<b>DBA scripts, database security, interface design, data visualization, data mining and Nosql databases.</b>	<b>Rupa Abhishek Siddhartha</b>	<b>25%</b>

## **Purpose**

The project deals with the management of the school database and contains all the details of faculty, students and courses. This document contains creating of tables, DBA scripts, performance tuning (indexing and parallelism) respectively.

## **Narrative**

Schools have students coming from different places and it is crucial for every school to maintain all the records of their students. Every student has a unique ID number that is stored in the database which has all the details of that particular student.

It is not mandatory that all the students pursue same course. So, it needs to maintain all the data related to the course work and class timings of each student.

In order for the students to register for their classes and check for the availability of different courses and faculty that matches their schedule they can access it from here anytime.

Every course would be handled by different professor and few courses may have multiple professors. All the schedules would be available in the database for every student to access it so that they know when their classes are and who would be handling it.

During their class work students can lend books from the library or can purchase them outside but every professor would suggest a different author for every course according to their easy of understanding all the suggested books and authors of the books are specified here.

## **Different Entities with individual Records**

1. Department Details
2. Student Details
3. Course Details
4. Faculty Details
5. Book Details
6. Session Details

## **Entities Attributes**

### **1) Department Details**

ID (Department ID – primary key)  
Department name  
Department contact number  
Department email

### **2) Student Details**

ID (primary key)  
First Name  
Last Name  
Contact Number  
Email  
Address  
State  
City  
Zip  
Department ID (Foreign key from department)  
Course ID (Foreign key from course)

### **3) Course Details**

ID (primary key)  
Course name  
Course length (number of days)  
Course type (Online/Offline)  
Department id (foreign key from department table)

### **4) Faculty Details**

ID (primary key)  
First name  
Last Name  
Designation  
Department id (foreign key from department table)  
Course id (foreign key from course table)

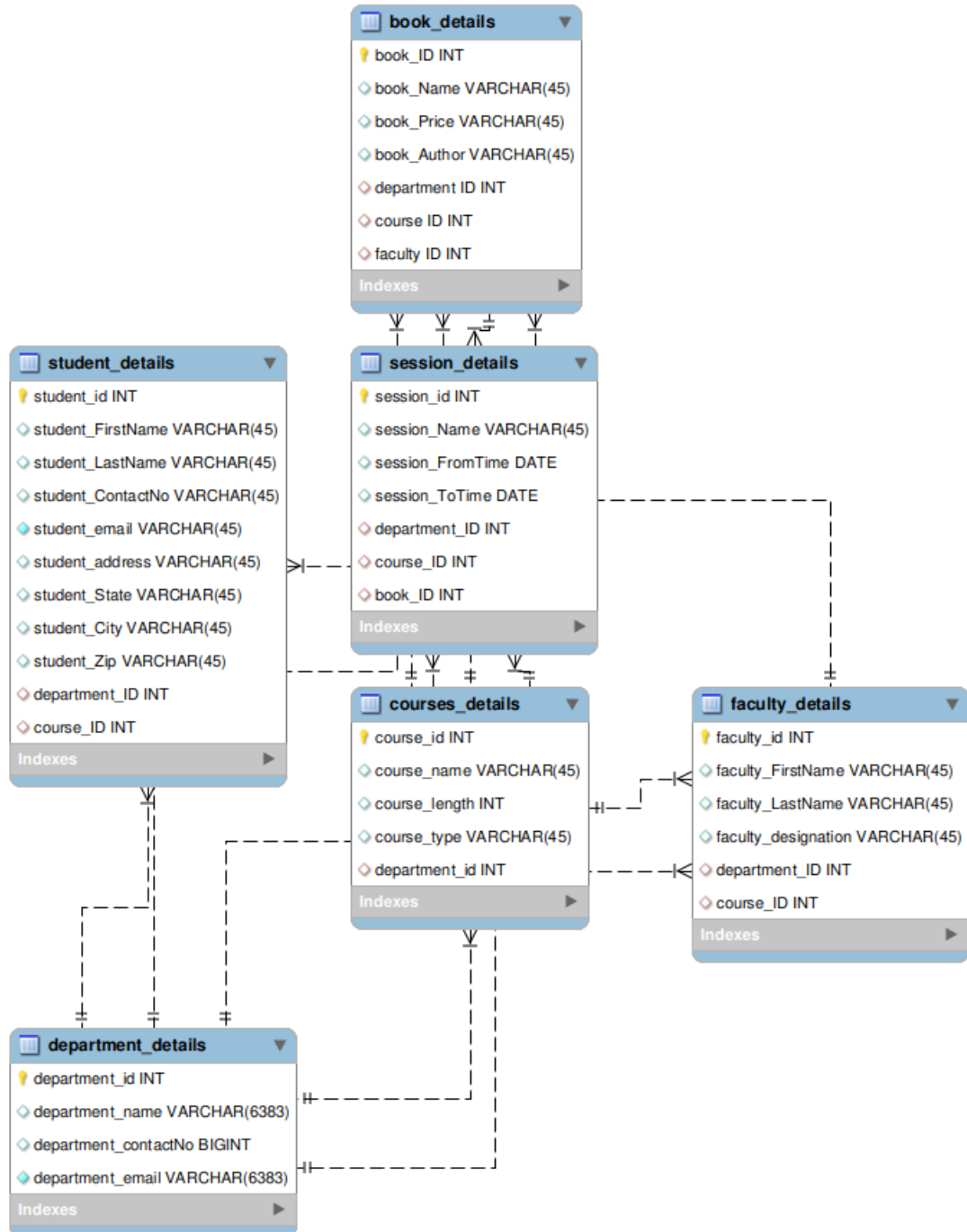
## **5) Book Details**

ID (primary key)  
Book name  
Book price  
Book author  
Department id (foreign key from department table)  
Course id (foreign key from course table)  
Faculty id (foreign key from faculty table)

## **6) Session Details: -**

ID (primary key)  
Name  
From time  
To time  
Department id (foreign key from department table)  
Course id (foreign key from course table)  
Book id (foreign key from book table)

## Entity Relationship Diagram



## Table Views

**Department Details:** This table contains all details about each department such as department ID (Primary key), department name, department contact number, department emailed.

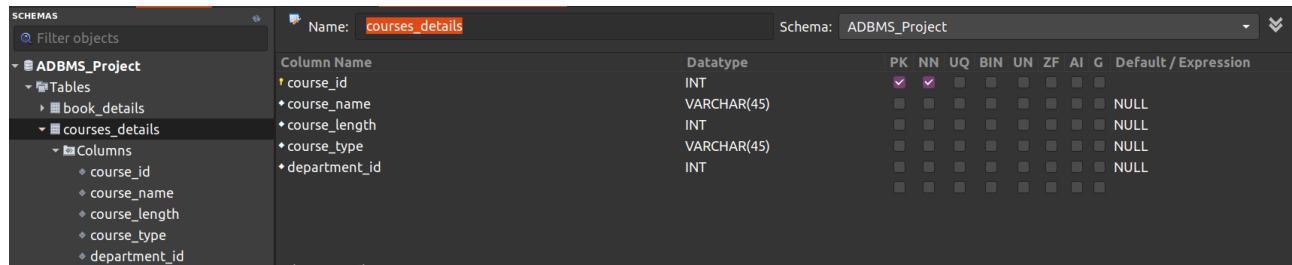
Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	G	Default / Expression
department_id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
department_name	VARCHAR(6383)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
department_contactNo	BIGINT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
department_email	VARCHAR(6383)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

**Student Details:** This table contains all the details of a student such as student ID (Primary key), First Name, Last Name, Contact Number, Email, Address, State, City, Zip, Department ID (Foreign Key from Department), Course ID (Foreign Key from Course table)

Column	Type	Default Value	Nulls	Character	Collation	Privileges	Extra	Comments
course_ID	int		YES			select,insert,update,references		
department_id	int		YES			select,insert,update,references		
student_addr	varchar(45)		YES	utf8mb4	utf8mb4_0900_ai_ci	select,insert,update,references		
student_City	varchar(45)		YES	utf8mb4	utf8mb4_0900_ai_ci	select,insert,update,references		
student_Conl	varchar(45)		YES	utf8mb4	utf8mb4_0900_ai_ci	select,insert,update,references		
student_ema	varchar(45)		NO	utf8mb4	utf8mb4_0900_ai_ci	select,insert,update,references		
student_First	varchar(45)		YES	utf8mb4	utf8mb4_0900_ai_ci	select,insert,update,references		
student_id	int		NO			select,insert,update,references		
student_Last	varchar(45)		YES	utf8mb4	utf8mb4_0900_ai_ci	select,insert,update,references		
student_Stat	varchar(45)		YES	utf8mb4	utf8mb4_0900_ai_ci	select,insert,update,references		
student_Zip	varchar(45)		YES	utf8mb4	utf8mb4_0900_ai_ci	select,insert,update,references		



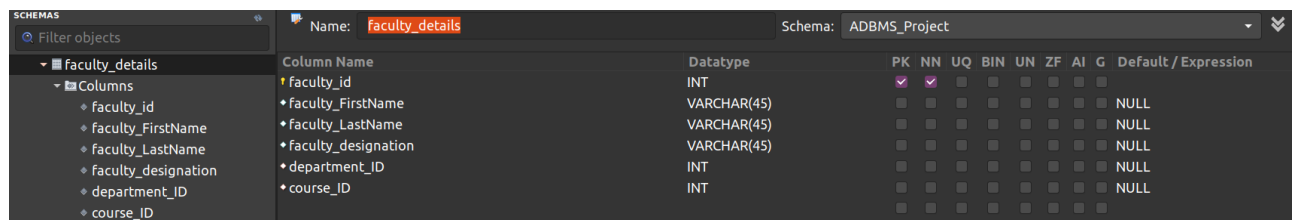
**Course Details:** This table has information about all the courses that are available such as course id, course name, duration of the course, course type (Online/Offline) and department ID (Foreign key from department table).



The screenshot shows a database schema tool interface. On the left, a tree view shows the 'ADBMS\_Project' schema with tables 'book\_details', 'courses\_details', and 'department\_details'. The 'courses\_details' table is selected. The main pane shows the table's structure with columns: course\_id (INT, PK, NN), course\_name (VARCHAR(45), NN), course\_length (INT, NN), course\_type (VARCHAR(45), NN), and department\_id (INT, NN). The 'course\_id' column is marked as the primary key.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	G	Default / Expression
course_id	INT	✓	✓							
course_name	VARCHAR(45)		✓							NULL
course_length	INT		✓							NULL
course_type	VARCHAR(45)		✓							NULL
department_id	INT		✓							NULL

**Faculty Details:** This table contains information of all the available faculty for the course like id (Primary key), First name, Last name, Designation, Department id (Foreign key from department table), Course Id (Foreign key from Course table)



The screenshot shows a database schema tool interface. On the left, a tree view shows the 'ADBMS\_Project' schema with tables 'book\_details', 'courses\_details', and 'faculty\_details'. The 'faculty\_details' table is selected. The main pane shows the table's structure with columns: faculty\_id (INT, PK, NN), faculty\_FirstName (VARCHAR(45), NN), faculty\_LastName (VARCHAR(45), NN), faculty\_designation (VARCHAR(45), NN), department\_ID (INT, NN), and course\_ID (INT, NN). The 'faculty\_id' column is marked as the primary key.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	G	Default / Expression
faculty_id	INT	✓	✓							
faculty_FirstName	VARCHAR(45)		✓							NULL
faculty_LastName	VARCHAR(45)		✓							NULL
faculty_designation	VARCHAR(45)		✓							NULL
department_ID	INT		✓							NULL
course_ID	INT		✓							NULL

**Book Details:** This table has the details about all the books that are available for the course like Book ID (Primary key), Book Name, Book Price, Author of the book, Department ID (Foreign key from department table), Faculty ID (Foreign key from faculty table), Course ID (Foreign key from course table).

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	G	Default / Expression
book_ID	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
book_Name	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
book_Price	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
book_Author	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
department_ID	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
course_ID	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
faculty_ID	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

**Session Details:** This table has information about the session available under the professor like session id, Name, From time, To time, Department ID (Foreign key from department table), Course ID (Foreign key from course table), Book ID (Foreign key from course table).

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	G	Default / Expression
session_id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
session_Name	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
session_FromTime	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
session_ToTime	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
department_ID	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
course_ID	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
book_ID	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

## **Data Integrity**

To maintain consistency of the data through the life cycle of data, integrity constraints are enforced. The constraints can either be at a column level or a table level. Some of the most common constraints are:

1. **NOT NULL** – Prevents a column from having a NULL value.
2. **PRIMARY KEY** – Uniquely identifies each row or record in table.
3. **FOREIGN KEY** – Uniquely identifies a column that references a PRIMARY KEY in another table.
4. **UNIQUE** – Prevents a column from having duplicate values.
5. **CHECK** – Checks for values that satisfy a specific condition as defined by the user.

Below are some of the constraints that we have enforced in our data base design

### 1. Table containing Department details

```
CREATE TABLE "ADBMS_Project"."department_details"
("department_id" NUMBER(*,0) NOT NULL,
"department_name" VARCHAR2(6383 BYTE),
"department_contactNo" NUMBER(10,0),
"department_email" VARCHAR2(6383 BYTE),
PRIMARY KEY ("department_id")
)
```

### 2. Table containing Student details

```
CREATE TABLE "ADBMS_Project"."student_details"
("sudent_id" NUMBER(*,0) NOT NULL,
"student_Firstname" VARCHAR2(50 BYTE),
"student_Lastname" VARCHAR2(50 BYTE),
"student_ContactNo" NUMBER(10,0),
"student_email" VARCHAR2(200 BYTE),
"student_address" VARCHAR(100 BYTE),
"student_State" CHAR(10 BYTE),
"student_City" VARCHAR2(20 BYTE),
"student_Zip" NUMBER(10,2),
"department_ID" NUMBER(*,0) NOT NULL,
"course_ID" NUMBER(*,0) NOTNULL
```

```
PRIMARY KEY ("student_id")  
)
```

### 3. Table containing Course details

```
CREATE TABLE "ADBMS_Project"."course_details"  
("course_id" NUMBER(*,0) NOT NULL,  
"course_name" VARCHAR2(45 BYTE),  
"course_length" NUMBER(10,2),  
"course_type" VARCHAR2(45  
BYTES),  
"department_id" NUMBER(*,0),  
PRIMARY KEY ("course_id")  
)
```

### 4. Table containing Faculty details

```
CREATE TABLE "ADBMS_Project"."faculty_details"  
("faculty_id" NUMBER(*,0),  
"faculty_Firstname" VARCHAR2(45 BYTES) NOT NULL,  
"faculty_Lastname" VARCHAR2(45 BYTE),  
"faculty_designation" VARCHAR2(45 BYTE),  
"department_id" NUMBER(5,2),  
"course_id" NUMBER(*,0),  
PRIMARY KEY ("faculty_id")  
)
```

### 5. Table containing Book Information

```
CREATE TABLE "ADBMS_Project"."book_details"  
("book_ID" NUMBER(*,0) NOT NULL,  
"book_Name" VARCHAR(45 BYTES),  
"book_Price" VARCHAR(45 BYTES),  
"book_Author" VARCHAR(45 BYTES),  
department ID NUMBER(*,0) NOT NULL,  
course ID NUMBER(*,0) NOT NULL,  
faculty ID NUMBER(*,0) NOT NULL,  
PRIMARY KEY ("book_ID")  
)
```

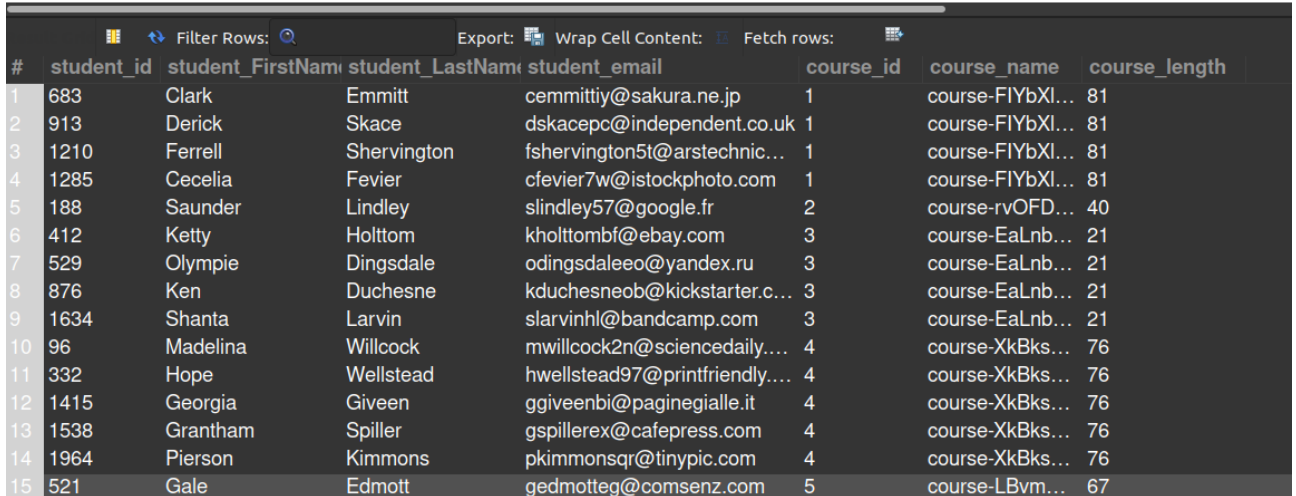
6. Table containing Session details

```
CREATE TABLE "ADBMS_Project"."session_details"  
  ("session_id" NUMBER(*,0) NOT NULL,  
  "session_Name" VARCHAR2(50 BYTE),  
  "session_FromTime" DATE,  
  "session_ToTime" DATE,  
  "department_ID" NUMBER(*,0),  
  "course_ID" NUMBER(*,0),  
  Book_ID NUMBER(*,0),  
  PRIMARY KEY ("session_id")  
)
```

# QUERIES

## 1. All the students for a given course ID

```
1 • select s.student_id,s.student_FirstName,s.student_LastName,s.student_email
2   ,c.course_id,c.course_name, c.course_length
3   from student_details s , courses_details c where c.course_id = s.course_ID;
```

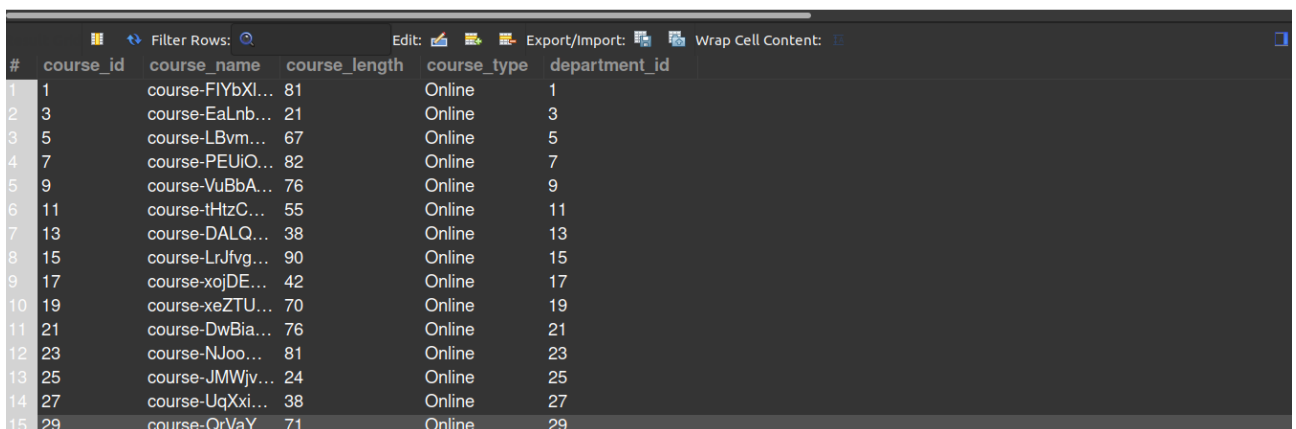


The screenshot shows a database application interface with a table of student details. The table has columns: student\_id, student\_FirstName, student\_LastName, student\_email, course\_id, course\_name, and course\_length. The data is displayed in 15 rows, numbered 1 to 15. The interface includes a search bar, a filter icon, and an export button.

#	student_id	student_FirstName	student_LastName	student_email	course_id	course_name	course_length
1	683	Clark	Emmitt	cemmittiy@sakura.ne.jp	1	course-FIYbXI...	81
2	913	Derick	Skace	dskacepc@independent.co.uk	1	course-FIYbXI...	81
3	1210	Ferrell	Shervington	fshervington5t@arstechnic...	1	course-FIYbXI...	81
4	1285	Cecelia	Fevier	cfevier7w@istockphoto.com	1	course-FIYbXI...	81
5	188	Saunder	Lindley	slindley57@google.fr	2	course-rvOFD...	40
6	412	Ketty	Holttom	kholttombf@ebay.com	3	course-EaLnb...	21
7	529	Olympie	Dingsdale	odingsdaleeo@yandex.ru	3	course-EaLnb...	21
8	876	Ken	Duchesne	kduchesneob@kickstarter.c...	3	course-EaLnb...	21
9	1634	Shanta	Larvin	slarvinhl@bandcamp.com	3	course-EaLnb...	21
10	96	Madelina	Willcock	mwillcock2n@sciencedaily....	4	course-XkBks...	76
11	332	Hope	Wellstead	hwellstead97@printfriendly....	4	course-XkBks...	76
12	1415	Georgia	Giveen	ggiveenbi@paginegialle.it	4	course-XkBks...	76
13	1538	Grantham	Spiller	gspillerex@cafepress.com	4	course-XkBks...	76
14	1964	Pierson	Kimmons	pkimmonsqr@tinypic.com	4	course-XkBks...	76
15	521	Gale	Edmott	gedmotteg@comsenz.com	5	course-LBvm...	67

## 2. Fetch the courses that are conducted “Online”

```
1 • select * from courses_details c where lower( c.course_type ) = "Online";
```



The screenshot shows a database application interface with a table of online courses. The table has columns: course\_id, course\_name, course\_length, course\_type, and department\_id. The data is displayed in 15 rows, numbered 1 to 15. The interface includes a search bar, a filter icon, and an export button.

#	course_id	course_name	course_length	course_type	department_id
1	1	course-FIYbXI...	81	Online	1
2	3	course-EaLnb...	21	Online	3
3	5	course-LBvm...	67	Online	5
4	7	course-PEUiO...	82	Online	7
5	9	course-VuBbA...	76	Online	9
6	11	course-tHtzC...	55	Online	11
7	13	course-DALQ...	38	Online	13
8	15	course-LrJfvg...	90	Online	15
9	17	course-xojDE...	42	Online	17
10	19	course-xeZTU...	70	Online	19
11	21	course-DwBia...	76	Online	21
12	23	course-NJoo...	81	Online	23
13	25	course-JMWjv...	24	Online	25
14	27	course-UqXxi...	38	Online	27
15	29	course-QrVaY...	71	Online	29

### 3. Get the faculty details with department and course taught by the faculty

```
1 • select f.faculty_id,f.faculty_FirstName,f.faculty_designation,
2 d.department_name, d.department_email,
3 c.course_name,c.course_type
4 from faculty_details f, department_details d, courses_details c
5 where f.department_ID = d.department_id and f.course_ID = c.course_id;
```

#	faculty_id	faculty_FirstName	faculty_designation	department_name	department_email	course_name	course_type
1	379	Agathe	Professor	department-qwbzViokduyb	abenuc@qwbzViokduyb.com	course-BboyUpUaGHAU	Online
2	235	Clemmie	Professor	department-qwbzViokduyb	abenuc@qwbzViokduyb.com	course-DKbnINmHJhtX	Online
3	120	Neddie	Professor	department-qwbzViokduyb	abenuc@qwbzViokduyb.com	course-kpQkWTQjCaQo	Offline
4	988	Cristen	Professor	department-jqkfpjUXRPwY	abuaj@jqkfpjUXRPwY.com	course-DfKvkuKQboej	Online
5	104	Alica	Professor	department-PISuJkErllIB	acesop@PISuJkErllIB.com	course-avOHmwGLLZpb	Online
6	466	Claudio	Professor	department-PISuJkErllIB	acesop@PISuJkErllIB.com	course-GKFDAmkqYwnl	Offline
7	860	Felita	Professor	department-dxSyOWKpSinh	acuen@dxSyOWKpSinh.com	course-edKTGVxMClwk	Offline
8	673	Tannev	Professor	department-dxSyOWKpSinh	acuen@dxSyOWKpSinh.com	course-mtzsGkRQEpJj	Offline
9	257	Sergio	Professor	department-dxSyOWKpSinh	acuen@dxSyOWKpSinh.com	course-sERYTvEnRpbq	Offline
10	948	Cristiano	Professor	department-dxSyOWKpSinh	acuen@dxSyOWKpSinh.com	course-WPONZcIDgcsM	Offline
11	643	Kalli	Professor	department-pyXtLJtjsrIU	ad@pyXtLJtjsrIU.com	course-yPTDGmSDsJcn	Online
12	529	Jan	Professor	department-VFYyrCDJgDjR	afi@VFYyrCDJgDjR.com	course-mgDcXfvOLroH	Offline
13	252	Harlan	Professor	department-VFYyrCDJgDjR	afi@VFYyrCDJgDjR.com	course-yISpxhhjCbH	Offline
14	790	Bobina	Professor	department-VFYyrCDJgDjR	afi@VFYyrCDJgDjR.com	course-zABmntjRBpMe	Online
15	992	Alden	Professor	department-YuMxWRzHagaE	afuvo@YuMxWRzHagaE.com	course-ozCIngeVnTDr	Offline

### 4. Get the student details who enrolled into courses of the respective departments

```
1 • select c.course_id,c.course_name,c.course_type,
2 s.student_FirstName,s.student_LastName,s.student_email, d.department_name from
3 courses_details c, student_details s, department_details d
4 where s.course_ID = c.course_id and d.department_id = s.department_ID;
```

#	course_id	course_name	course_type	student_FirstName	student_LastName	student_email	department_name
1	1	course-FIYbXINauktd	Online	Clark	Emmitt	cemmittiy@sakura.ne.jp	department-gGMKzBRFWJFu
2	1	course-FIYbXINauktd	Online	Derick	Skace	dskacepc@independent.co.uk	department-zumKbNPEabw
3	1	course-FIYbXINauktd	Online	Ferrell	Shervington	fshervington5t@arstechnic...	department-ClfSuSvDoxJvk
4	1	course-FIYbXINauktd	Online	Cecelia	Fevier	cfevier7w@istockphoto.com	department-llhRKxAmgFTf
5	2	course-rvOFDqGlubxl	Offline	Saunder	Lindley	slindley57@google.fr	department-BbGRXOalUsXY
6	3	course-EaLnbWwRnxSk	Online	Ketty	Holttom	kholttombf@ebay.com	department-dIAyPTKdDsQQ
7	3	course-EaLnbWwRnxSk	Online	Olympie	Dingsdale	odingsdaleeo@yandex.ru	department-bbOSXDPAhUHI
8	3	course-EaLnbWwRnxSk	Online	Ken	Duchesne	kduchesneob@kickstarter.c...	department-QDgPedjEUOPz
9	3	course-EaLnbWwRnxSk	Online	Shanta	Larvin	slarvinhl@bandcamp.com	department-iAsMSGFNslxU
10	4	course-XkBkszXMLAJU	Offline	Madelina	Willcock	mwillcock2n@sciencedaily....	department-ldSgsRCKvPKv

## 5. Fetch sessions done in a particular course conducted by a department

```
1 • select s.session_Name, s.session_FromTime as session_FromDate, s.session_ToTime as session_ToDate,
2 c.course_name,c.course_length,c.course_type,
3 d.department_name
4 from session_details s, courses_details c, department_details d where s.course_ID = c.course_id
5 and d.department_id = s.department_ID;
```

#	session_Name	session_FromDate	session_ToDate	course_name	course_len	course_type	department_name
1	Customer-focused attitude-ori...	2021-04-18	2021-09-17	course-XInnniZkWSiO	82	Offline	department-LQPZDRCiKmhI
2	Customer-focused responsive ...	2021-06-13	2021-01-25	course-nVDLRILGUaDV	71	Offline	department-QDgPedjEUOpZ
3	Fundamental responsive utilis...	2020-05-06	2020-03-30	course-yYCulwFwWjC	24	Offline	department-dEdCzCremkco
4	Object-based demand-driven ...	2020-03-24	2020-08-29	course-VsLTdkTeTXF	31	Offline	department-PjQhEFsshvXA
5	Open-source stable neural-net	2020-07-07	2021-03-18	course-DHXwewSoFVPO	34	Online	department-BybkuOpeGgsO
6	Persevering tertiary functiona...	2021-02-08	2020-08-11	course-DtjdlMGjImS	52	Offline	department-fLOBgjWXfmrM
7	Profit-focused real-time standa...	2020-03-30	2020-12-28	course-olZklZwMHwaD	62	Online	department-lklWavdjxRof
8	Reduced object-oriented instal...	2021-06-02	2020-10-12	course-MYyoesdQtFdF	80	Online	department-ILqgIECSwRjV
9	Right-sized optimal conglomer...	2021-08-20	2020-11-07	course-quQeEzPqcMgh	80	Online	department-jNVAdHMntHTA
10	Stand-alone uniform flexibility	2021-01-06	2021-01-22	course-HaEDNXkXXcqt	48	Offline	department-OORyknmxhyTb

## 6. Fetch Faculty from department

```
1 • select d.department_id, d.department_name,d.department_email,
2 f.faculty_id,f.faculty_FirstName,f.faculty_LastName,f.faculty_designation
3 from department_details d, faculty_details f
4 where d.department_id = f.department_ID;
```

#	department_id	department_name	department_email	faculty_id	faculty_FirstName	faculty_LastName	faculty_designation
1	333	department-olpgegKkjXxc	aroeljoh@olpgegKkjXxc.com	1	Ted	Podbury	Professor
2	180	department-KTDboSfzVRRRA	hihhulbow@KTDboSfzVRRRA...	2	Loleta	Buessen	Professor
3	400	department-JtJpTJXUTotH	poloro@JtJpTJXUTotH.com	3	Beverlee	Cona	Professor
4	481	department-VFKvhtWRlyao	pa@VFKvhtWRlyao.com	4	Asa	Wallman	Professor
5	469	department-UPBSVSrpmngB	isiwohreg@UPBSVSrpmngB...	5	Sibeal	Chipchase	Professor
6	265	department-aowFbFXcPMES	goipe@aowFbFXcPMES.com	6	Linet	Aucourte	Professor
7	59	department-mTrHgcRtmBtk	edfe@mTrHgcRtmBtk.com	7	Conrade	Whitland	Professor
8	83	department-wlDmkbfkFeka	rusop@wlDmkbfkFeka.com	8	Karyn	Haggeth	Professor
9	58	department-oRvThAfKwAa	livafovis@oRvThAfKwAa.com	9	Trenton	Nunnerley	Professor
10	17	department-mOSmoBTeUV...	nivo@mOSmoBTeUVPP.com	10	Keriann	Teresse	Professor



## 7. Display the books available in offered courses and sessions

```

1 • select b.books_Name,b.books_Author,b.books_Price,
2   c.course_id,c.course_name,c.course_type,
3   s.session_id,s.session_FromTime as session_FromDate, s.session_ToTime as session_ToDate
4   from books_details b, courses_details c, session_details s
5   where b.course_id = c.course_id and s.book_ID = b.books_id;

```

#	books_Name	books_Author	books_Price	course_id	course_name	course_type	session_id	session_FromDate	session_ToDate
1	Carphophis sp.	Derry Garnham	445	190	course-rQaSncOboNnN	Offline	1	2020-07-07	2021-03-18
2	Limnocolax flavirostra	Dylan Pachta	550	35	course-iAKfOIsaxyYO	Online	2	2020-03-24	2020-08-29
3	Anastomus oscitans	Nataniel Weavill	421	353	course-wmqeRjSnjYEj	Online	3	2021-08-20	2020-11-07
4	Vulpes vulpes	Annora Izard	732	409	course-zRHegnIGEozW	Online	4	2021-06-02	2020-10-12
5	Sitta canadensis	Corrianne Bilve...	873	245	course-XRvpzIoWuCDA	Online	5	2021-04-18	2021-09-17
6	Falco peregrinus	Xenia Petrillo	316	415	course-yZuNREZVqhlz	Online	6	2020-05-06	2020-03-30
7	Ephippiorhynchus s...	Maison Hamm	792	219	course-aSeWcTEyYXQi	Online	7	2021-01-06	2021-01-22
8	Larus dominicanus	Jeniffer Zarfai	624	281	course-QuxxPiUXUWXW	Online	9	2020-03-30	2020-12-28
9	Bettongia penicillata	Cullin Moraleda	244	389	course-SFfsMFZRAwMN	Online	10	2021-06-13	2021-01-25
10	Buteo galapagoensis	Hasheem Airey	203	293	course-kSFMqGnqXcqy	Online	11	2021-02-08	2020-08-11

## Performance Tuning

### Indexing

An index is used to increase the overall performance of queries. Indexing does this by reducing the data pages that has to be visited or scanned every time a query is run.

When we create index, by default the primary key creates a clustered index. In SQL Server, a clustered index determines the physical order of data in a table. There can be only one clustered index per table.

### Query:

```
1 • SELECT * FROM student_details WHERE student_id BETWEEN 1 AND 11;
```

#	student_id	student_FirstName	student_LastName	student_ContactNo	student_email	student_address
1	1	Bogey	Giral	+86-835-296-8647	bgiral0@typepad.com	322 Londonderry Hill
2	2	Maighidin	Dominin	+351-761-740-7855	mdominin1@facebook.com	723 Sommers Circle
3	3	Carlin	Popping	+66-785-328-8251	cpopping2@hubpages.com	60072 Mosinee Drive
4	4	Carmelle	Kiendl	+353-769-362-7552	ckiendl3@ftc.gov	8702 Holy Cross Road
5	5	Mordecai	Oury	+251-134-519-3694	moury4@illinois.edu	02 Emmet Point
6	6	Malena	Wakeford	+66-968-368-5665	mwakeford5@dailyml.co.uk	4 Leroy Center
7	7	Cecil	Topliss	+1-967-980-6551	ctopliss6@w3.org	36 Longview Point
8	8	Euell	Ditt	+86-998-307-9860	edit7@xing.com	55 Esker Avenue
9	9	Estella	McDell	+86-135-429-8596	emcdell8@theglobeandmail.com	7 Stang Plaza
10	10	Clarice	Huston	+86-428-891-9135	chuston9@a8.net	9500 Shasta Pass
11	11	Valli	Dewberry	+291-450-939-1047	vdewberry@utexas.edu	879 Cordelia Way

### Parallelism:

```
1 • SELECT /*+PARALLEL(4)*/ a.faculty_id, a.faculty_FirstName, a.faculty_LastName, a.faculty_designation,  
2 COUNT(b.department_id)  
3 FROM faculty_details a, department_details b  
4 WHERE a.department_ID = b.department_id  
5 GROUP BY a.faculty_id,a.faculty_FirstName,a.faculty_LastName;
```

Tabular Explain ▼							
id	select_type	table	partitions	type	possible_keys	key	key_len ref
1	SIMPLE	a		ALL	department_id_faculty_idx		
1	SIMPLE	b		eq_ref	PRIMARY	PRIMARY	4 ADBMS_Project.a.de

## Optimizer Mode:

Optimizer mode is used to choose better execution plans for poorly written queries.

The available optimizer modes for development environments are all\_rows, first\_rows, first\_rows(1|10|100|1000), choose, rules.

Below we have used first\_rows optimizer\_mode to demonstrate the example:-  
FIRST\_ROWS (1|10|100|1000) Gets the first n rows faster. This is good for applications that routinely display partial results to users such as paging data to a user in a web application.

## Query:

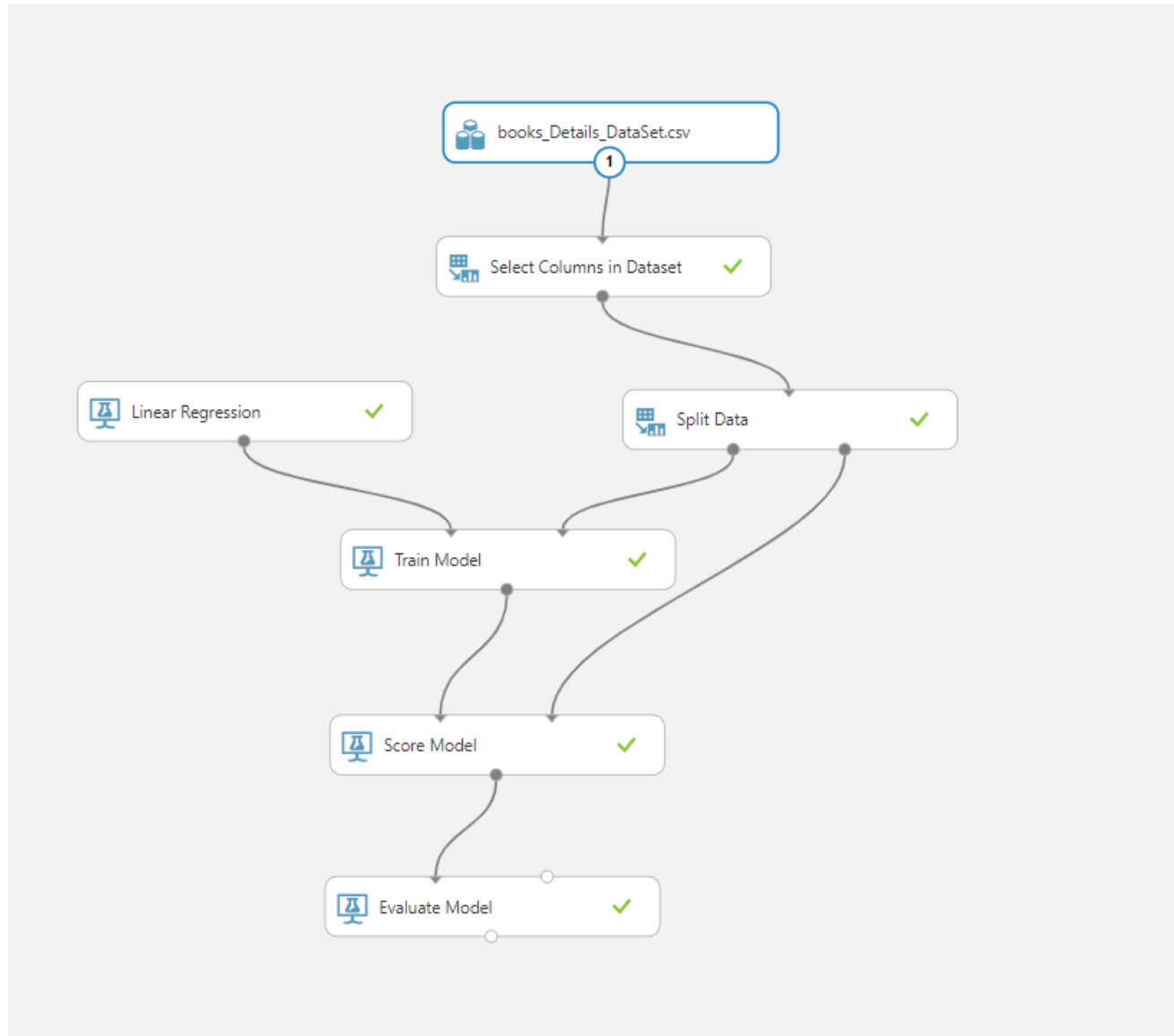
1 •

SELECT /\*+FIRST\_ROWS(10)\*/ FROM student\_details where student\_id BETWEEN 1031 AND 1040;

#	student_id	student_FirstName	student_LastName	student_ContactNo	student_email	student_address
1	1031	Sherill	Spivey	+7-254-278-3904	sspiveyu@mit.edu	28 Lake View Park
2	1032	Suzy	Hammonds	+351-365-424-5722	shammondsv@gravatar.com	63 Glacier Hill Lane
3	1033	Koenraad	Duffan	+218-520-206-8454	kduffanw@thetimes.co.uk	5 Di Loreto Plaza
4	1034	Justine	Motherwell	+46-499-633-8756	jmotherwell@nasa.gov	4945 Lien Avenue
5	1035	Bete	Jaxon	+52-788-982-4177	bjaxony@newyorker.com	6438 Hanson Street
6	1036	Humberto	Stockall	+98-785-891-3395	hstockallz@webnode.com	39730 Graedel Circle
7	1037	Sandro	Tott	+506-500-548-1403	stott10@indiatimes.com	108 Pennsylvania Circle
8	1038	Crisie	Mence	+27-537-778-7145	cmence11@chron.com	522 Dryden Point
9	1039	Audrie	Kingsmill	+55-460-192-3599	akingsmill12@addtoany.com	548 Hollow Ridge Lane
10	1040	Daisey	Yaldren	+81-945-574-8226	dyaldren13@mapquest.com	17 Delaware Hill
11	NULL	NULL	NULL	NULL	NULL	NULL

## Data Visualization

### AZURE Machine Learning Model




## Data Set:

Experiment created on 11/25/2021 » books\_Details\_DataSet.csv » dataset

rows  
1499

columns  
7

	books_id	books_Name	books_Price	books_Author	department_ID	course_ID	faculty_ID
view as							
	1	Felis chaus	544	Eddy Alleyn	194	126	143
	2	Buteo galapagoensis	448	Toby Norman	65	74	44
	3	Notechis semmiannulatus	548	Hurlee Moncreif	488	449	189
	4	Bettongia penicillata	244	Cullin Moraleda	121	389	747
	5	Rhea americana	457	Bibby Cadman	380	471	424
	6	Gyps bengalensis	302	Madalyn Lusgdin	103	319	513
	7	Phalaropus lobatus	859	Imojean MacSweeney	457	334	935
	8	Cordylus giganteus	350	Hattie Pawlyn	389	329	597
	9	Bubalornis niger	503	Ashlee Aubin	432	492	980
	10	Semnopithecus entellus	426	Booth Gernier	373	432	251
	11	Chelodina longicollis	887	Leonore Drover	63	30	852
	12	Lamprotornis nitens	659	Cyrus Conaboy	136	181	62
	13	Rangifer tarandus	706	Antons Kolakovic	234	344	695
	14	Cynictis penicillata	769	Meris Tuminelli	104	147	994
	15	Paroaria gularis	243	Tansy Feeney	26	404	135
	16	Phascogale calura	410	Kynthia Golding	380	108	811
	17	Zalophus californicus	228	Willi Gomersal	367	158	407
	18	Felis caracal	397	Janine Palumbo	164	138	967
	19	Ara ararauna	565	Page Rantoull	108	34	959
	20	Pteropus rufus	940	Ogden Luttger	452	290	162
	21	Sciurus vulgaris	969	Shelbi Newband	200	187	669
	22	Eurocephalus anguitimens	858	Cathrin Ivanyushkin	463	301	103
	23	Sciurus niger	334	Aaron Blagburn	147	355	117

## Department ID vs Book name:

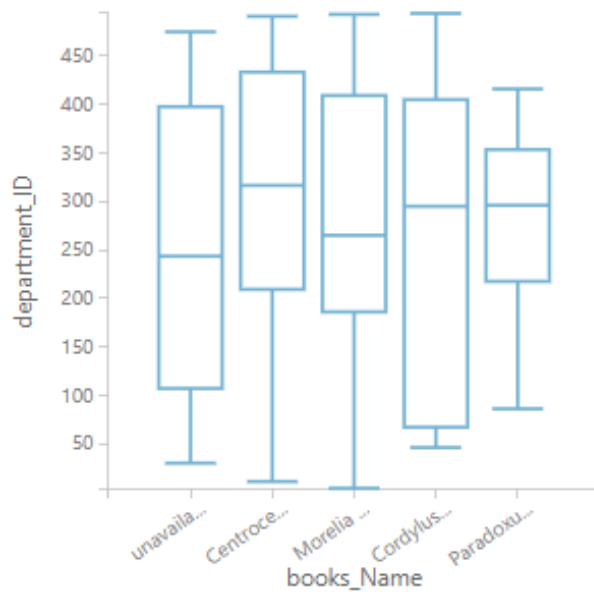
Median	247
Min	1
Max	500
Standard Deviation	146.0177
Unique Values	479
Missing Values	0
Feature Type	Numeric Feature

### Visualizations

department\_ID

MultiboxPlot

compare to books\_Name



☐ department\_ID log scale

categories

5

## Metrics and Error Histogram

Experiment created on 11/25/2021 » Evaluate Model » Evaluation results

### Metrics

Mean Absolute Error	388.90736
Root Mean Squared Error	468.217545
Relative Absolute Error	1.082597
Relative Squared Error	1.255866
Coefficient of Determination	-0.255866

### Error Histogram

