Introduction-Case Study

Kathmandu College is the newly opened College of Science and Technology in Kathmandu, Bafal. This college has a Student,staff,Teacher,Library and Sport Department.Numbers of student,teacher,staff,Sport items and book items are present there.

Problem Statement:

This college is newly opened so it needs the database system to keep the record of financial transactions.

Requirements

This college requires a solution that would enable them to add Student,Teacher and staff.As well as college can open their account and keep the record of transactions. The solution must have the following requirements:

i. Student : The solution must allow add student's Information in system .

ii. Teacher:The solution must allow add teacher's Information in system

iii.Staff :The solution must allow add Staff's Information in system

iv.Account : The solution must allow to add salaries for student and fee for student.Likewise, maintain the record of the salary withdraw and fee pay for student respectively.

v .Sports Item: The Solution must allow to add sport items in system and keep record of return and issued.

vi.BookItem: The solution must allow adding books in the system.

vii. Fine Amount:The solution must allow add fine in the system.

Solution

To meet the requirements of Kathmandu College ,we propose a system with the following features:

1 .Student Table: The system will allow to students and their information in the system

2. Staff Table :The system will allow to Staffs and their information in the system.

3. Teacher Table : The system will allow to Teacher and their information in the system

4 Student Account Table :The system will allow to record fee that student pay as well as remaining fee,and date when student pay

5 Teacher Account Table: The system will record the salary of individuals teacher along With their salary credit and remaining salary.

6. Staff Account Table : The system will record the salary of individuals staff along With their salary credit and remaining salary.

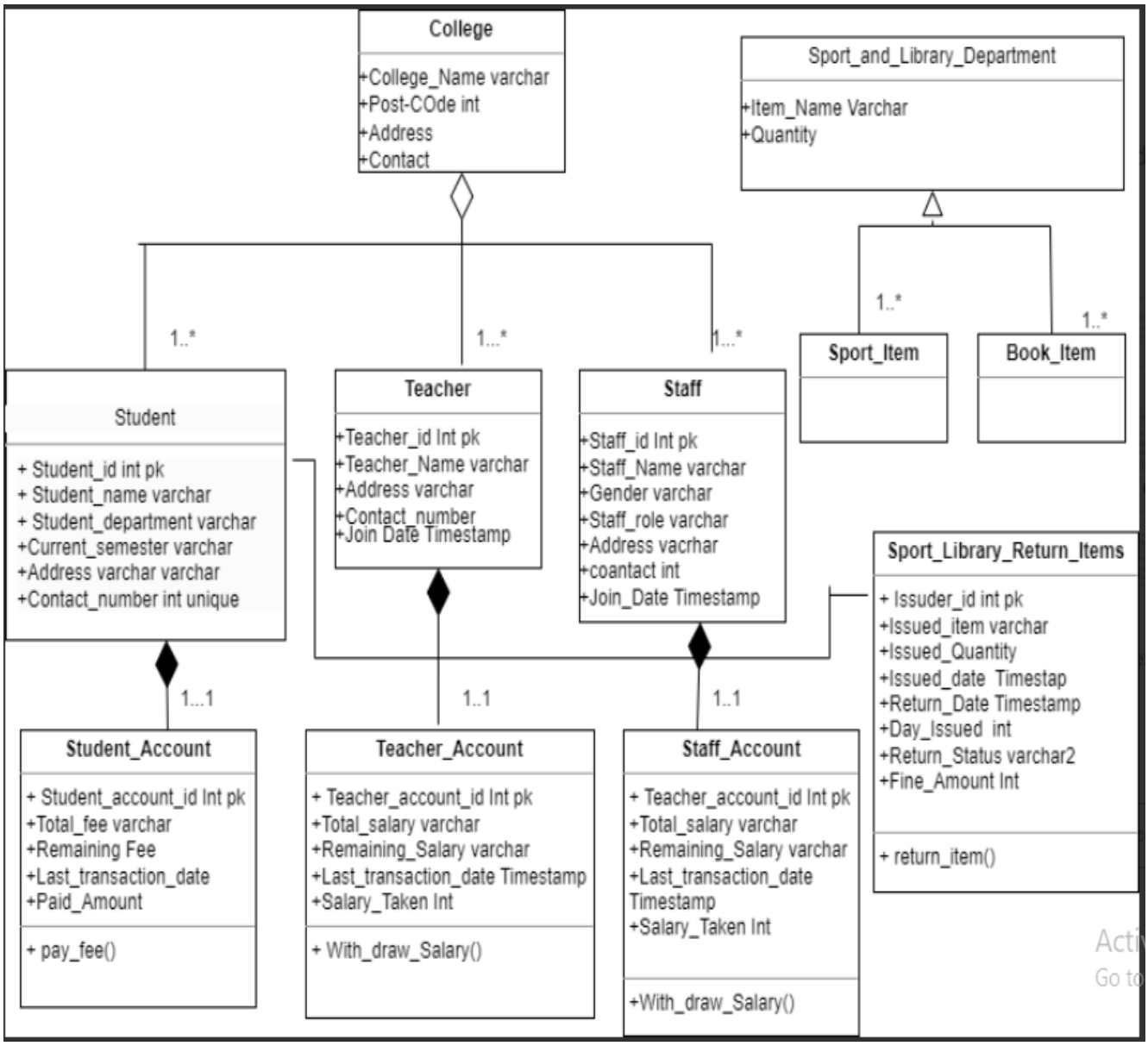
7 Book_items Table: The system will allow to record the books in the system

8 Sports_item Table: The system will allow to record the sport_items in the system

9 Issued_and_return_item_Table:The system will allow to keep issued and return records along with fine amount.

Task One :UML Class Diagram

UML class Diagram of CollegeDataBase



Task Two:

Demonstration of SQL Code and its working.

a. Drop Tables and Sequence

Query:

```
drop table College;

drop table Student_Account;
drop table Student;
drop sequence student_id_sequence;

drop table  Teacher_account;
drop table Teacher;
drop sequence teacher_id_sequence;

drop table Staff_account;
drop table Staff;
drop sequence staff_id_sequence;
```

Output

```
Table dropped.

Table dropped.

Table dropped.

Sequence dropped.

Table dropped.

Table dropped.

Sequence dropped.

Table dropped.

Table dropped.

Sequence dropped.
```

b. Drop Triggers

Query:

```
1  drop trigger student_insert_trigger;
2  drop trigger teacher_id_trigger;
3  drop trigger staff_id_trigger;
```

Output

```
Trigger dropped.

Trigger dropped.

Trigger dropped.
```

c. Drop Procedure

Query

```
18
19  drop procedure salary_withdraw_teacher;
20  drop procedure pay_fee;
21  drop procedure deposite_salary_to_teacher_account;
22
23  drop procedure salary_withdraw_staff;
24  drop procedure deposite_salary_to_staff_account
25
26
```

Output:

```
Procedure  dropped.

Procedure  dropped.

Procedure  dropped.

Procedure  dropped.

Procedure  dropped.
```

c.Drop Type and Table

Query:

```
25
26  drop table Sport_Items;
27  drop type Sport_Department;
```

Output:

```
Table  dropped.

Type  dropped.
```

Query:

```
28
29  drop table  Books_Items;
30  drop type Library_Department;
31  drop type Sport_and_Library_Department;
```

Output:

```
Table  dropped.

Type  dropped.

Type  dropped.
```

Query:

```
33
34  drop  table Sport_Items__Issued_and_Return_Item;
35  drop type Sport_Department__Issued_and_Return_Item;
36  drop type Sport_and_Library_Department_Issued_and_Return_Item;
```

Output:

```
Table  dropped.

Type  dropped.

Type  dropped.
```

Creating Table, sequence,procedure and Trigger:

Query:

```
41  create Table College(
42      college_Name varchar(50),
43      post_code int,
44      address varchar(50),
45      contact int
46  );
47
48  create table Student(
49      student_id varchar(5) primary key,
50      student_name varchar(30),
51      student_department varchar(30),
52      student_current_semester varchar(30),
53      student_address varchar(30),
54      student_contact_number int unique
55  );
56  create Table Student_Account(
57      student_account_id varchar(5) primary key,
58      total_fee varchar(10),
59      remaining_fee varchar(10),
60      last_transaction_date Timestamp,
61      paid_amount varchar(10)
```

Output:

```
Table created.

Table created.

Table created.
```

Query:

```
66
67  create sequence student_id_sequence
68        start with 1
69        Increment by 1;
70
71  CREATE OR REPLACE TRIGGER student_insert_trigger
72  BEFORE INSERT ON Student
73  FOR EACH ROW
74  DECLARE
75      v_student_id VARCHAR(5);
76  BEGIN
77      -- Generate the student_id
78      v_student_id := 's' || student_id_sequence.NEXTVAL;
79      -- Insert the generated student_id into Student_Account with default values
80      INSERT INTO Student_Account (student_account_id, total_fee, remaining_fee, last_transaction_date, paid_amount)
81      VALUES (v_student_id, '8000', '8000', NULL, NULL);
82
83     :NEW.student_id := v_student_id;
84  END;
85  /
```

Output:

```
Sequence created.

Trigger created.
```

Query:

```
92
93  create table Teacher(
94        teacher_id varchar(5) primary key,
95        teacher_name varchar(25),
96        teacher_address varchar(20),
97        contact_number  int unique,
98        join_date Timestamp
99    );
100
101 create table Teacher_account(
102       teacher_account_id varchar(5) primary key,
103       total_salary varchar(5),
104       remaining_salary varchar(5),
105       last_transaction_date Timestamp,
106       salary_taken varchar(30)
107   );
108 create sequence teacher_id_sequence
109         start with 1
110         Increment by 1;
```

Output:

```
Table created.

Table created.

Sequence created.
```

Query:

```
119
120 create or replace trigger teacher_id_trigger
121 before insert on Teacher
122 for each row
123 declare
124     v_teacher_id varchar(5);
125 begin
126     v_teacher_id := 't' || teacher_id_sequence.NEXTVAL;
127     Insert into Teacher_account(teacher_account_id,total_salary,remaining_salary,last_transaction_date,salary_taken)
128     values (v_teacher_id,'0','0',NULL,'0');
129     :NEW.teacher_id := v_teacher_id;
130 End;
131 /
```

Output:

```
Trigger created.
```

Query:

```
create table Staff(
      staff_id varchar(5) primary key,
      staff_name varchar(25),
      gender  varchar(10),
      staff_role varchar(20),
      staff_address varchar(20),
      contact_number  int unique,
      join_date Timestamp
);
create table Staff_account(
      staff_account_id varchar(5) primary key,
      total_salary varchar(5),
      remaining_salary varchar(5),
      last_transaction_date Timestamp,
      salary_taken varchar(30)
);
create sequence staff_id_sequence
       start with 1
       Increment by 1;

```

Output:

```
Table created.

Table created.

Sequence created.
```

Query:

```
168
169
170 create or replace trigger staff_id_trigger
171 before insert on Staff
172 for each row
173 declare
174     v_staff_id varchar(5);
175 begin
176     v_staff_id := 'st' || staff_id_sequence.NEXTVAL;
177     Insert into Staff_account(staff_account_id,total_salary,remaining_salary,last_transaction_date,salary_taken)values(v_staff_id,'0','0',NULL,'0');
178     :NEW.staff_id := v_staff_id;
179 End;
180 /
181
```

Output:

```
Trigger created.
```

Query:

```
Create or Replace Procedure pay_fee(
      p_student_id in varchar2,
      p_amount_paid in varchar2
)
As
     v_total_fee varchar2(10);
     v_remaining_fee varchar2(10);
Begin
     select total_fee,remaining_fee into v_total_fee,v_remaining_fee
     from Student_Account
     where student_account_id = p_student_id;

     v_remaining_fee := To_Char(To_number(v_total_fee)-To_number(p_amount_paid));
     update Student_Account
     set remaining_fee = v_remaining_fee,
         last_transaction_date = SYSTIMESTAMP,
         paid_amount = p_amount_paid
    where  student_account_id = p_student_id;
End pay_fee;
/
```

Output:

```
Procedure created.
```

Query:

```
231  --------Procedure for Admin to deposite Salary of Teacher to Teacher_Account at Monthend----------
232  create or replace procedure deposite_salary_to_teacher_account(
233      p_teacher_id in varchar2,
234      p_salary_to_deposite in varchar2
235  )
236  as
237     v_total_salary varchar2(10);
238     v_remaining_salary varchar2(10);
239  begin
240     select total_salary,remaining_salary into v_total_salary,V_remaining_salary
241     from Teacher_account
242     where teacher_account_id = p_teacher_id;
243
244     v_total_salary := To_char(To_number(V_remaining_salary) + To_number(p_salary_to_deposite));
245     v_remaining_salary := v_total_salary;
246     update Teacher_account
247     set total_salary = v_total_salary,
248         remaining_salary = v_remaining_salary
249     where
250         teacher_account_id = p_teacher_id;
251
252  End deposite_salary_to_teacher_account;
253  /
```

Output:

Query:

```
263
264 create or replace procedure deposite_salary_to_staff_account(
265     p_staff_id in varchar2,
266     p_salary_to_deposite in varchar2
267 )
268 as
269     v_total_salary varchar2(10);
270     v_remaining_salary varchar2(10);
271
272 begin
273     select total_salary,remaining_salary into v_total_salary,V_remaining_salary
274     from Staff_account
275     where staff_account_id = p_staff_id;
276
277     v_total_salary := To_char(To_number(V_remaining_salary) + To_number(p_salary_to_deposite));
278     v_remaining_salary := v_total_salary;
279     update Staff_account
280     set total_salary = v_total_salary,
281         remaining_salary = v_remaining_salary
282     where
283         staff_account_id = p_staff_id;
284
285 End deposite_salary_to_staff_account;
286 /
```

Output:

Query:

```
---------Procedure to withdraw salary for Teacher-----------
create or replace procedure salary_withdraw_teacher(
    p_teacher_id in varchar2,
    p_salary_withdraw in varchar2
)
as
   v_total_salary varchar2(10);
   v_remaining_salary varchar2(10);
begin
   select total_salary,remaining_salary into v_total_salary,v_remaining_salary
   from Teacher_account
   where teacher_account_id = p_teacher_id;
   If To_number(v_remaining_salary) >= To_number(p_salary_withdraw) Then

       v_remaining_salary := To_Char(To_number(v_remaining_salary)-To_number(p_salary_withdraw))
       update Teacher_account
       set remaining_salary = v_remaining_salary,
          last_transaction_date = SYSTIMESTAMP,
          salary_taken = p_salary_withdraw
       where teacher_account_id = p_teacher_id;
       DBMS_OUTPUT.PUT_LINE('Salary Withdraw successfully');
   else
       DBMS_OUTPUT.PUT_LINE('sorry!!!InSufficient Balance.');
   end If;

End salary_withdraw_teacher;
```

Output:

Query:

```
create or replace procedure salary_withdraw_staff(
    p_staff_id in varchar2,
    p_salary_withdraw in varchar2
)
as
   v_total_salary varchar2(10);
   v_remaining_salary varchar2(10);
begin
   select total_salary,remaining_salary into v_total_salary,v_remaining_salary
   from Staff_account
   where staff_account_id = p_staff_id;
   If To_number(v_remaining_salary) >= To_number(p_salary_withdraw) Then

       v_remaining_salary := To_Char(To_number(v_remaining_salary)-To_number(p_salary_withdraw));
       update Staff_account
       set remaining_salary = v_remaining_salary,
          last_transaction_date = SYSTIMESTAMP,
          salary_taken = p_salary_withdraw
       where staff_account_id = p_staff_id;
       DBMS_OUTPUT.PUT_LINE('Salary Withdraw successfully');
   else
       DBMS_OUTPUT.PUT_LINE('sorry!!!InSufficient Balance.');
   end If;

End salary_withdraw_staff;
/
```

Output:

```
Procedure created.
```

Query:

```
376 create or replace type Sport_and_Library_Department as object(
377     Item_Name varchar(30),
378     Quantity int
379 )Not Final;
380
381 create or replace type Sport_Department under Sport_and_Library_Department();
382 create table Sport_Items of Sport_Department;
```

Output:

```
Type created.
```

```
Type created.
```

```
Table created.
```

Query:

```
388

389  create or replace type Library_Department under Sport_and_Library_Department();
390  create table Books_Items of Library_Department;
```

Output:

```
Type created.
```

```
Table created.
```

Query:

```
395
396
397 create or replace type Sport_and_Library_Department_Issued_and_Return_Item as object(
398     Issuder_id varchar(20),
399     Issed_item varchar(20),
400     Issed_quantity varchar(5) ,
401     Issued_date timestamp,
402     Return_date timestamp,
403     Days_Issued number,
404     Return_stattus varchar2(3),
405     Fine_Amount varchar2(5)
406 )Not Final;
407 create or replace type Sport_Department__Issued_and_Return_Item under Sport_and_Library_Department_Issued_and_Return_Item();
408 create table Sport_Items__Issued_and_Return_Item of Sport_Department__Issued_and_Return_Item;
```

Output:

```
Type created.
```

```
Type created.
```

```
Table created.
```

Query:

```
409  -- drop table Sport_Items__Issued_and_Return_Item;
410  create or replace procedure update_sport_items(
411      p_issuder_id varchar2,
412      p_issed_item varchar2,
413      p_issed_quantity varchar2,
414      p_issed_date timestamp
415  )
416  as
417  begin
418     update Sport_Items
419     set Quantity = Quantity - to_number(p_issed_quantity)
420     where Item_Name = p_issed_item;
421  end;
422  /
```

Output:

```
Procedure created.
```

Query:

```
423
424 create or replace trigger  Sport_Department_Issued_and_Return_Item_Trigger
425 After insert on Sport_Items_Issued_and_Return_Item
426 for each row
427 Begin
428     update_sport_items(:new.Issuder_id,:new.Issed_item, :new.Issed_quantity,:new.Issued_date);
429 End;
430 /
```

Output:

```
Trigger created.
```

Query:

```
CREATE OR REPLACE PROCEDURE return_sport_items(
 p_issuder_id VARCHAR2,
   p_issed_item VARCHAR2,
   p_issed_quantity VARCHAR2,
   p_return_date TIMESTAMP
)
AS
 v_total_return_quantity VARCHAR2(10);
 v_quantity VARCHAR2(10);
 v_days_diff NUMBER;
 v_issued_date TIMESTAMP;
BEGIN
 -- Use a cursor to handle multiple rows in the SELECT
 FOR c IN (
  SELECT Issed_quantity, Issued_date
  FROM Sport_Items__Issued_and_Return_Item
  WHERE Issuder_id = p_issuder_id
    AND Issed_item = p_issed_item
 )
 LOOP
  v_total_return_quantity := c.Issed_quantity;
  v_issued_date := c.Issued_date;
 END LOOP;

 -- Use a cursor to handle multiple rows in the SELECT
 FOR c IN (
  SELECT Quantity
  FROM Sport_Items
  WHERE Item_Name = p_issed_item
 )
 LOOP
  v_quantity := c.Quantity;
 END LOOP;

 -- Check if returned quantity is less than or equal to issued quantity
 IF TO_NUMBER(p_issed_quantity) = TO_NUMBER(v_total_return_quantity) THEN
```

```
  -- Update Sport_Items only if the condition is met
  UPDATE Sport_Items
  SET Quantity = TO_NUMBER(v_quantity) + TO_NUMBER(p_issed_quantity)
  WHERE Item_Name = p_issed_item;
  v_days_diff := EXTRACT(DAY FROM (p_return_date - v_issued_date));

  -- Update existing row in Sport_Items__Issued_and_Return_Item
  UPDATE Sport_Items__Issued_and_Return_Item
  SET Return_date = p_return_date,
      Days_Issued = v_days_diff,
      Return_stattus = 'YES',
      Fine_Amount =
       CASE
         WHEN v_days_diff > 10 AND v_days_diff <= 20 THEN 50
         WHEN v_days_diff > 20 AND v_days_diff <= 30 THEN 100
         ELSE 500
       END
  WHERE Issuder_id = p_issuder_id
    AND Issed_item = p_issed_item;

  DBMS_OUTPUT.PUT_LINE('Update');
 ELSE
  DBMS_OUTPUT.PUT_LINE('Sorry!!! Issued and Return quantity do not match.');
 END IF;
END return_sport_items;
/
```

Output:

```
Procedure created.
```

Inserting Values in Tables:

Query:

```
83
84  Insert into Student(student_name,student_department,student_current_semester,student_address,student_contact_number) values
85      ('Ram Poudel','IT','3rd','Kathmandu',9845697859);
86  Insert into Student(student_name,student_department,student_current_semester,student_address,student_contact_number) values
87      ('Prem kc','Engineering','2nd','Bhatapur',984569459723);
88  Insert into Student(student_name,student_department,student_current_semester,student_address,student_contact_number) values
89      ('Pratiba Magar','IT','3rd','Biratnagar',9845697547);
90  select * from Student_Account;
91
```

Output:

STUDENT_ID	STUDENT_NAME	STUDENT_DEPARTMENT	STUDENT_CURRENT_SEMESTER	STUDENT_ADDRESS	STUDENT_CONTACT_NUMBER
s1	Ram Poudel	IT	3rd	Kathmandu	9845697859
s2	Prem kc	Engineering	2nd	Bhatapur	984569459723
s3	Pratiba Magar	IT	3rd	Biratnagar	9845697547

Download CSV

Query:

```
67
68  CREATE OR REPLACE TRIGGER student_insert_trigger
69  BEFORE INSERT ON Student
70  FOR EACH ROW
71  DECLARE
72      v_student_id VARCHAR(5);
73  BEGIN
74      -- Generate the student_id
75      v_student_id := 's' || student_id_sequence.NEXTVAL;
76      -- Insert the generated student_id into Student_Account with default values
77      INSERT INTO Student_Account (student_account_id, total_fee, remaining_fee, last_transaction_date, paid_amount)
78      VALUES (v_student_id, '8000', '8000', NULL, NULL);
79
80     :NEW.student_id := v_student_id;
81  END;
82  /
```

select * from Student_Account;

Output:

STUDENT_ACCOUNT_ID	TOTAL_FEE	REMAINING_FEE	LAST_TRANSACTION_DATE	PAID_AMOUNT
s1	8000	8000	-	-
s2	8000	8000	-	-
s3	8000	8000	-	-

Download CSV

Query:

```
133
134  Insert into Teacher(teacher_name,teacher_address,contact_number,join_date) values
135      ('Bikas Dhakal','Kathmandu',9845678945,Timestamp '2022-1-15 14:30:00' );
136  Insert into Teacher(teacher_name,teacher_address,contact_number,join_date) values
137      ('Mina Kumari','Patan',9845678874,Timestamp '2022-2-15 2:30:00' );
138  Insert into Teacher(teacher_name,teacher_address,contact_number,join_date) values
139      ('Ram K.c','Baglung',9845678569,Timestamp '2022-3-30 1:25:00' );
140  Insert into Teacher(teacher_name,teacher_address,contact_number,join_date) values
141      ('Sita Pradhan','Kathmandu',9845678666,Timestamp '2022-3-31 1:3:00' );
142  Select * from Teacher;
```

Output:

TEACHER_ID	TEACHER_NAME	TEACHER_ADDRESS	CONTACT_NUMBER	JOIN_DATE
t1	Bikas Dhakal	Kathmandu	9845678945	15-JAN-22 02.30.00.000000 PM
t2	Mina Kumari	Patan	9845678874	15-FEB-22 02.30.00.000000 AM
t3	Ram K.c	Baglung	9845678569	30-MAR-22 01.25.00.000000 AM
t4	Sita Pradhan	Kathmandu	9845678666	31-MAR-22 01.03.00.000000 AM

Download CSV

Query:

```
Select * from Teacher_account;
```

Output:

TEACHER_ACCOUNT_ID	TOTAL_SALARY	REMAINING_SALARY	LAST_TRANSACTION_DATE	SALARY_TAKEN
t1	0	0	-	0
t2	0	0	-	0
t3	0	0	-	0
t4	0	0	-	0

Download CSV

Query:

```
185
186  Insert into Staff(staff_name,gender,staff_role,staff_address,contact_number,join_date) values
187      ('Shuba Tamang','Female','Cleaning','Kathmandu',98465987623,Timestamp '2022-1-15 14:30:00');
188  Insert into Staff(staff_name,gender,staff_role,staff_address,contact_number,join_date) values
189      ('Sita Poudel','Female','Cooking','Sagarmatha',98465987625,Timestamp '2022-1-16 10:30:00');
190  Insert into Staff(staff_name,gender,staff_role,staff_address,contact_number,join_date) values
191      ('Binita Kc','Female','Helper','Hetauda',98465987627,Timestamp '2022-5-15 11:30:00');
192  Insert into Staff(staff_name,gender,staff_role,staff_address,contact_number,join_date) values
193      ('Rita Magar','Female','Cleaning','Pokhara',98465987648,Timestamp '2022-6-15 12:30:00');
194  Insert into Staff(staff_name,gender,staff_role,staff_address,contact_number,join_date) values
195      ('Ravi Magar','Male','Driver','Pokhara',98465987665415,Timestamp '2022-6-14 12:30:00');
196  Insert into Staff(staff_name,gender,staff_role,staff_address,contact_number,join_date) values
197      ('Kopila Magar','Female','Cleaning','Birgunj',98461651,Timestamp '2022-6-17 12:30:00');
198  Insert into Staff(staff_name,gender,staff_role,staff_address,contact_number,join_date) values
199      ('Shiva Puri','Male','Lab assistance','Sagarmatha',98466215448,Timestamp '2022-6-20 12:30:00');
200  Insert into Staff(staff_name,gender,staff_role,staff_address,contact_number,join_date) values
201      ('Salin Thakuri','Male','Data Administator','Kathmandu',984659448,Timestamp '2022-6-26 12:30:00');
202  Insert into Staff(staff_name,gender,staff_role,staff_address,contact_number,join_date) values
203      ('Kabita Luitel','Female','Cleaning','Pokhara',984659854588,Timestamp '2022-6-27 12:30:00');
204
```

Output:

STAFF_ID	STAFF_NAME	GENDER	STAFF_ROLE	STAFF_ADDRESS	CONTACT_NUMBER	JOIN_DATE
st1	Shuba Tamang	Female	Cleaning	Kathmandu	98465987623	15-JAN-22 02.30.00.000000 PM
st2	Sita Poudel	Female	Cooking	Sagarmatha	98465987625	16-JAN-22 10.30.00.000000 AM
st3	Binita Kc	Female	Helper	Hetauda	98465987627	15-MAY-22 11.30.00.000000 AM
st4	Rita Magar	Female	Cleaning	Pokhara	98465987648	15-JUN-22 12.30.00.000000 PM
st5	Ravi Magar	Male	Driver	Pokhara	98465987665415	14-JUN-22 12.30.00.000000 PM
st6	Kopila Magar	Female	Cleaning	Birgunj	98461651	17-JUN-22 12.30.00.000000 PM
st7	Shiva Puri	Male	Lab assistance	Sagarmatha	98466215448	20-JUN-22 12.30.00.000000 PM
st8	Salin Thakuri	Male	Data Administator	Kathmandu	984659448	26-JUN-22 12.30.00.000000 PM
st9	Kabita Luitel	Female	Cleaning	Pokhara	984659854588	27-JUN-22 12.30.00.000000 PM

Query:

select * from Staff_account;

Table is created automatically when Staff is created because of Trigger and Table staff_id_trigger and Staff_Account respectively

Output:

STAFF_ACCOUNT_ID	TOTAL_SALARY	REMAINING_SALARY	LAST_TRANSACTION_DATE	SALARY_TAKEN
st1	0	0	-	0
st2	0	0	-	0
st3	0	0	-	0
st4	0	0	-	0
st5	0	0	-	0
st6	0	0	-	0
st7	0	0	-	0
st8	0	0	-	0
st9	0	0	-	0

Download CSV

Query:

```
237  Execute pay_fee('s1',3000);
238  Execute pay_fee('s2',5000);
239  Execute pay_fee('s3',2000);
240  select * from Student_Account;
241
```

Output:

STUDENT_ACCOUNT_ID	TOTAL_FEE	REMAINING_FEE	LAST_TRANSACTION_DATE	PAID_AMOUNT
s1	8000	5000	30-DEC-23 05.45.51.634011 PM	3000
s2	8000	3000	30-DEC-23 05.45.51.640247 PM	5000
s3	8000	6000	30-DEC-23 05.45.51.643235 PM	2000

Download CSV

Query:

```
269  Execute deposite_salary_to_teacher_account('t1',2000);
270  Execute deposite_salary_to_teacher_account('t2',3000);
271  Execute deposite_salary_to_teacher_account('t3',1500);
272  Execute deposite_salary_to_teacher_account('t4',1600);
273  select * from Teacher_account;
```

OutPut:

TEACHER_ACCOUNT_ID	TOTAL_SALARY	REMAINING_SALARY	LAST_TRANSACTION_DATE	SALARY_TAKEN
t1	2000	2000	-	0
t2	3000	3000	-	0
t3	1500	1500	-	0
t4	1600	1600	-	0

Download CSV

Query:

```
303
304   execute deposite_salary_to_staff_account('st1',800);
305   execute deposite_salary_to_staff_account('st2',900);
306   execute deposite_salary_to_staff_account('st3',1000);
307   execute deposite_salary_to_staff_account('st4',700);
308   execute deposite_salary_to_staff_account('st5',800);
309   execute deposite_salary_to_staff_account('st6',900);
310   execute deposite_salary_to_staff_account('st7',1000);
311   execute deposite_salary_to_staff_account('st8',700);
312   execute deposite_salary_to_staff_account('st9',700);
313   select * from Staff_account;
314
315
```

Output:

STAFF_ACCOUNT_ID	TOTAL_SALARY	REMAINING_SALARY	LAST_TRANSACTION_DATE	SALARY_TAKEN
st1	800	800	-	0
st2	900	900	-	0
st3	1000	1000	-	0
st4	700	700	-	0
st5	800	800	-	0
st6	900	900	-	0
st7	1000	1000	-	0
st8	700	700	-	0
st9	700	700	-	0

Download CSV

Query:

```
404 create or replace type Library_Department under Sport_and_Library_Department();
405 create table Books_Items of Library_Department;
406 Insert into Books_Items(Item_Name,Quantity) values ('C++', 50);
407 Insert into Books_Items(Item_Name,Quantity) values ('DSA', 50);
408 Insert into Books_Items(Item_Name,Quantity) values ('Python', 50);
409 Insert into Books_Items(Item_Name,Quantity) values ('Calculas', 50);
```

Output:

ITEM_NAME	QUANTITY
C++	50
DSA	50
Python	50
Calculas	50

Query:

```
396 create or replace type Sport_Department under Sport_and_Library_Department();
397 create table Sport_Items of Sport_Department;
398 Insert into Sport_Items(Item_Name,Quantity) values ('Football', 5);
399 Insert into Sport_Items(Item_Name,Quantity) values ('Basketball', 5);
400 Insert into Sport_Items(Item_Name,Quantity) values ('TT', 20);
401 Insert into Sport_Items(Item_Name,Quantity) values ('CriketSet', 2);
402 select * from Sport_Items;
403
```

Output:

ITEM_NAME	QUANTITY
Football	5
Basketball	5
TT	20
CriketSet	2

Download CSV

Query:

```
448
449 Insert into Sport_Items__Issued_and_Return_Item
450 (Issuder_id,Issed_item,Issed_quantity,Issued_date,Return_date,Days_Issued,Return_stattus,Fine_Amount)Values
451 ('s1','Football',3,To_TIMESTAMP('2022-09-01 12:00:00', 'YYYY-MM-DD HH24:MI:SS'),NULL,NULL,NULL,NULL);
452 Insert into Sport_Items__Issued_and_Return_Item
453 (Issuder_id,Issed_item,Issed_quantity,Issued_date,Return_date,Days_Issued,Return_stattus,Fine_Amount)Values
454 ('s2','TT',3,To_TIMESTAMP('2022-09-01 12:00:00', 'YYYY-MM-DD HH24:MI:SS'),NULL,NULL,NULL,NULL);
455 select * from Sport_Items__Issued_and_Return_Item;
```

Output:

ISSUDER_ID	ISSED_ITEM	ISSED_QUANTITY	ISSUED_DATE	RETURN_DATE	DAYS_ISSUED	RETURN_STATTUS	FINE_AMOUNT
s1	Football	3	01-SEP-22 12.00.00.000000 PM	-	-	-	-
s2	TT	3	01-SEP-22 12.00.00.000000 PM	-	-	-	-

Query:

Execute return_sport_items('s1','Football',3,To_TIMESTAMP('2022-10-01 12:00:00', 'YYYY-MM-DD HH24:MI:SS'));

Select * from return_sport_items;

Output:

ISSUDER_ID	ISSED_ITEM	ISSED_QUANTITY	ISSUED_DATE	RETURN_DATE	DAYS_ISSUED	RETURN_STATTUS	FINE_AMOUNT
s1	Football	3	01-SEP-22 12.00.00.000000 PM	01-OCT-22 12.00.00.000000 PM	30	YES	100
s2	TT	3	01-SEP-22 12.00.00.000000 PM	-	-	-	-

Download CSV

Descriptions:

The script initializes by dropping existing tables, sequences, types, procedures, and triggers associated with the CollegeDataBase. It then proceeds to create essential sequences for entities likeCollege,Student,Teacher, Staff, Teacher_Account,Staff_Account,Sport_Items,Books_Items, Sport_Items__Issued_and_Return_Item Tables .

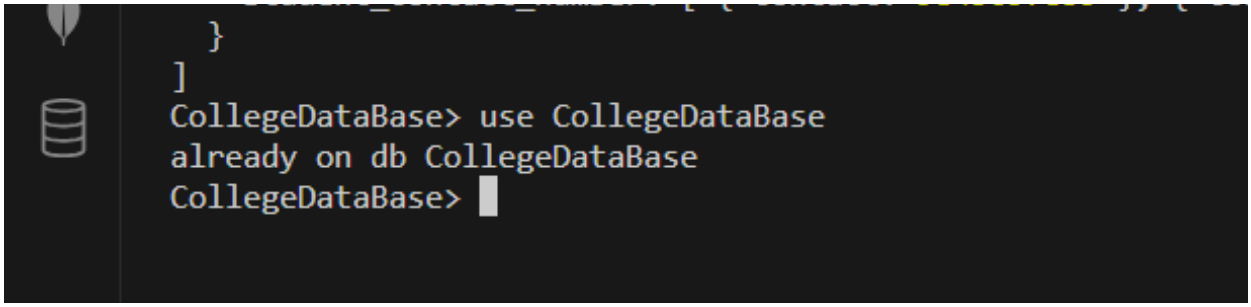
The Script defines tables for these entities,establishes triggers for generating unique IDs, and inserts sample data into the tables.Additionally,it creates procedures for handling financial transactions like fee payments and salary deposits and withdrawals for both teacher and staff members.

It also include types and tables for managing items in the Sport and Library Department,including issuance and return of items. Foreign key constraint are added to maintain referential integrity.The join operations are demonstrated with examples, and OLAP function are used to calculate the total salary sum groped by gender in staff table.

Task Three: Demonstration of MongoDB Scripts and its working

Using CollegeDataBase>
Query:

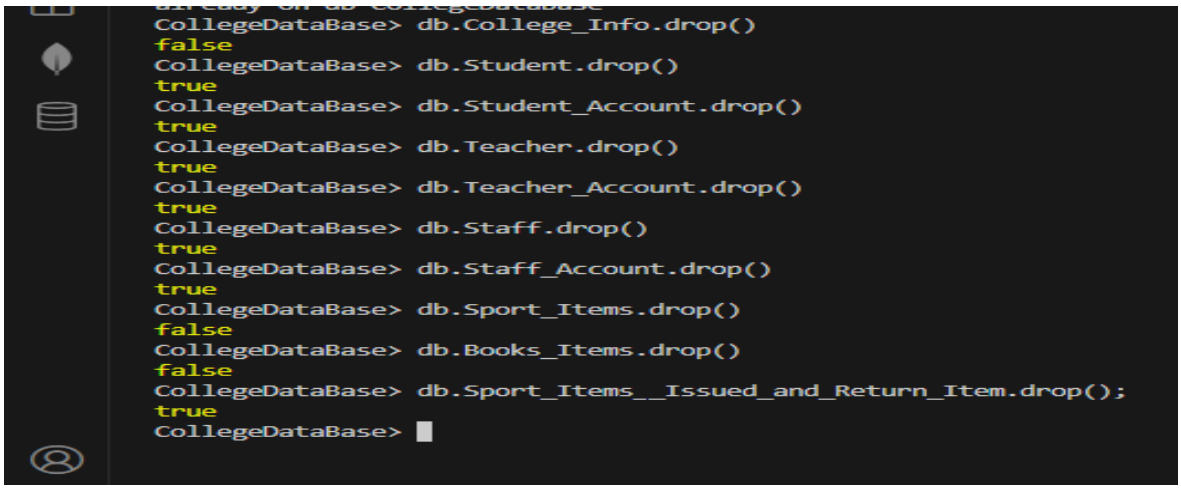
```
use CollegeDataBase
```



Query

```
db.College_Info.drop()
db.Student.drop()
db.Student_Account.drop()
db.Teacher.drop()
db.Teacher_Account.drop()
db.Staff.drop()
db.Staff_Account.drop()
db.Sport_Items.drop()
db.Books_Items.drop()
db.Sport_Items__Issued_and_Return_Item.drop();
```

Output:



Query:

```
db.createCollection("College_Info");
db.College_Info.insertOne(
    {
    college_Name :'Sagarmatha College of Science and Technology',
    post_code : 5256,
    address : 'Sanepa',
    contact : [
        {
            contact1:15156498
        },
        {
            contact2:151515412
        }
    ]
});
```

Output:

```
CollegeDataBase> db.createCollection("College_Info");
{ ok: 1 }
CollegeDataBase> db.College_Info.insertOne(
...     {
.....     college_Name :'Sagarmatha College of Science and Technology',
.....     post_code : 5256,
.....     address : 'Sanepa',
.....     contact : [
.....         {
.......             contact1:15156498
.......           },
.....         {
.......             contact2:151515412
.......           }
.....     ]
..... });
{
  acknowledged: true,
  insertedId: ObjectId("65906066cf9aa31201f0dc5e")
}
CollegeDataBase>
```

Query:

```
db.createCollection("Student");
db.Student.insertMany([
    {
      student_id : 's1',
      student_name: 'Ram Poudel',
      student_department: 'IT',
      student_current_semester: '3rd',
      student_address: 'Kathmandu',
      student_contact_number: [
        {
            contact:9845697859
        },
        {
            contact:9846456456
        }
      ]
    },
    {
      student_id : 's2',
      student_name: 'Prem kc',
      student_department: 'Engineering',
      student_current_semester: '2nd',
      student_address: 'Bhatapur',
      student_contact_number: [
        {
            contact:984569459723
        },
        {
            contact:9845694597141

        }
      ]
    },
    {
      student_id : 's3',
      student_name: 'Pratiba Magar',
      student_department: 'IT',
      student_current_semester: '3rd',
```

```
    student_address: 'Biratnagar',
    student_contact_number: [
      {
          contact:9845697547
      },
      {
          contact:9845697547
      }
    ]
  }
]);
```

Output:

```
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("65906171cf9aa31201f0dc5f"),
    '1': ObjectId("65906171cf9aa31201f0dc60"),
    '2': ObjectId("65906171cf9aa31201f0dc61")
  }
}
CollegeDataBase>
```

Query:

```
db.createCollection("Student_Account")
db.Student_Account.insertMany([
    {
        "STUDENT_ACCOUNT_ID": "s1",
        "TOTAL_FEE": 8000,
        "REMAINING_FEE": 5000,
        "LAST_TRANSACTION_DATE": "2023-12-30T04:18:13.953Z",
        "PAID_AMOUNT": 3000
      },
      {
        "STUDENT_ACCOUNT_ID": "s2",
        "TOTAL_FEE": 8000,
        "REMAINING_FEE": 3000,
        "LAST_TRANSACTION_DATE": "2023-12-30T04:18:57.018Z",
        "PAID_AMOUNT": 5000
      },
      {
        "STUDENT_ACCOUNT_ID": "s3",
        "TOTAL_FEE": 8000,
        "REMAINING_FEE": 6000,
        "LAST_TRANSACTION_DATE": "2023-12-30T04:19:37.804Z",
        "PAID_AMOUNT": 2000
      }
]);
```

Output:

```
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("6590622acf9aa31201f0dc62"),
    '1': ObjectId("6590622acf9aa31201f0dc63"),
    '2': ObjectId("6590622acf9aa31201f0dc64")
  }
}
CollegeDataBase>
```

Query:

```
db.createCollection("Teacher");
db.Teacher.insertMany([
    {
      teacher_id : 't1',
      teacher_name: 'Bikas Dhakal',
      teacher_address: 'Kathmandu',
      contact_number: [
        {
            contact1:9845678945
        },
        {
            contact2:9845678945

        }
      ],
      join_date: ISODate('2022-01-15T14:30:00.000Z')
    },
    {
      teacher_id : 't2',
      teacher_name: 'Mina Kumari',
      teacher_address: 'Patan',
      contact_number: 9845678874[
        {
            contact1:9845678874
        },
        {
            contact2:98456785615

        }
      ],
      join_date: ISODate('2022-02-15T02:30:00.000Z')
    },
    {
      teacher_id : 't3',
      teacher_name: 'Ram K.c',
      teacher_address: 'Baglung',
      contact_number: 9845678569[
        {
```

```
            contact1:9845678569
        },
        {
            contact2:9845678566

        }
    ],
    join_date: ISODate('2022-03-30T01:25:00.000Z')
  },
  {
    teacher_id : 't4',
    teacher_name: 'Sita Pradhan',
    teacher_address: 'Kathmandu',
    contact_number: 9845678666[
      {
          contact1:9845678666
      },
      {
          contact2:98456151545

      }
    ],
    join_date: ISODate('2022-03-31T01:03:00.000Z')
  }
]);
```

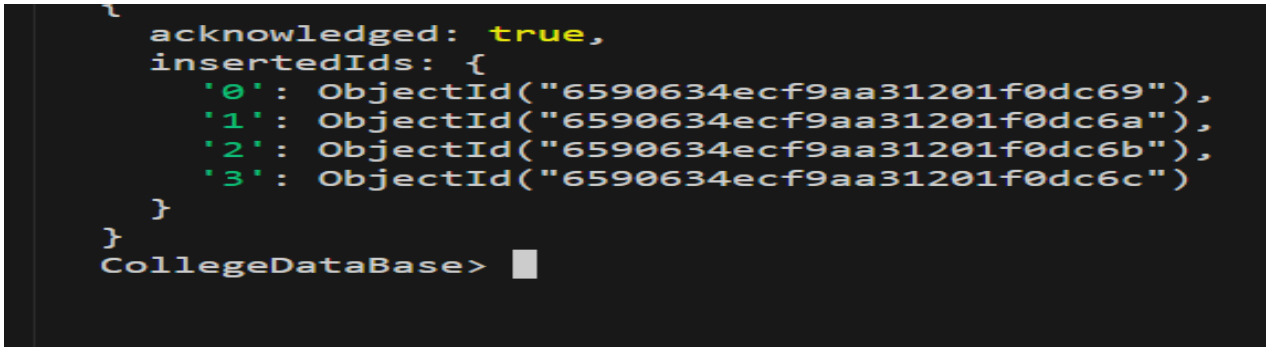
Output:

```
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("659062e6cf9aa31201f0dc65"),
    '1': ObjectId("659062e6cf9aa31201f0dc66"),
    '2': ObjectId("659062e6cf9aa31201f0dc67"),
    '3': ObjectId("659062e6cf9aa31201f0dc68")
  }
}
CollegeDataBase>
```

Query:

```
db.createCollection("Teacher_Account");
db.Teacher_Account.insertMany([
    {
        "TEACHER_ACCOUNT_ID": "t1",
        "TOTAL_SALARY": 2000,
        "REMAINING_SALARY": 1300,
        "LAST_TRANSACTION_DATE": "2023-12-30T04:47:34.259Z",
        "SALARY_TAKEN": 700
      },
      {
        "TEACHER_ACCOUNT_ID": "t2",
        "TOTAL_SALARY": 3000,
        "REMAINING_SALARY": 2500,
        "LAST_TRANSACTION_DATE": "2023-12-30T04:47:34.266Z",
        "SALARY_TAKEN": 500
      },
      {
        "TEACHER_ACCOUNT_ID": "t3",
        "TOTAL_SALARY": 1500,
        "REMAINING_SALARY": 900,
        "LAST_TRANSACTION_DATE": "2023-12-30T04:47:34.271Z",
        "SALARY_TAKEN": 600
      },
      {
        "TEACHER_ACCOUNT_ID": "t4",
        "TOTAL_SALARY": 1600,
        "REMAINING_SALARY": 700,
        "LAST_TRANSACTION_DATE": "2023-12-30T04:47:34.275Z",
        "SALARY_TAKEN": 900
      }

])
```

Output:



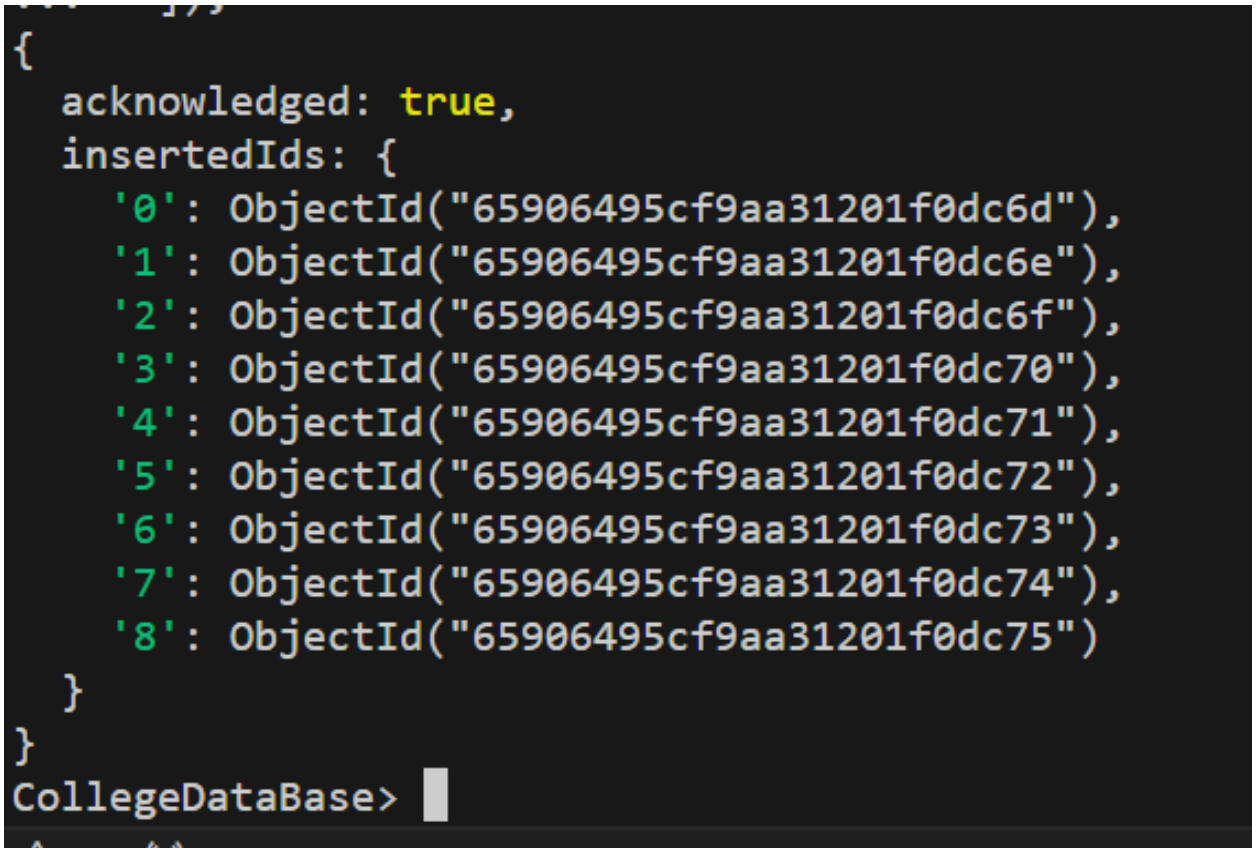
Query:

```
db.createCollection("Staff");
db.Staff.insertMany([
    {
    staff_id:'st1',
    staff_name: 'Shuba Tamang',
    gender:"Female",
    staff_role: 'Cleaning',
     staff_address: 'Kathmandu',
     contact_number: [
        {
            contact1:98465987623
        },
        {
            contact2:98456151454

        }
      ],
     join_date: new Date('2022-01-15T14:30:00')
     },

    { staff_id:'st2',
      staff_name: 'Sita Poudel',
      gender:"Female",
      staff_role: 'Cooking',
      staff_address: 'Sagarmatha',
      contact_number:[
        {
            contact1:98465987625
        },
        {
            contact2:9845615444

        }
      ],
      join_date: new Date('2022-01-16T10:30:00')
    },
    { staff_id:'st3',
      staff_name: 'Binita Kc',
      gender:"Female",
      staff_role: 'Helper',
      staff_address: 'Hetauda',
      contact_number:[
        {
            contact1:98465987627
        },
        {
            contact2:9845611215

        }
      ],
      join_date: new Date('2022-05-15T11:30:00')
    },
```

```
{
staff_id:'st4' ,
staff_name: 'Rita Magar',
gender:"Female",
staff_role: 'Cleaning',
staff_address: 'Pokhara',
contact_number: [
    {
        contact1:98465987648
    },
    {
        contact2:98456155545

    }
  ],
join_date: new Date('2022-06-15T12:30:00')
},
{
    staff_id: "st5",
    staff_name: "Ravi Magar",
    gender: "Male",
    staff_role: "Driver",
    staff_address: "Pokhara",
    contact_number:[
        {
            contact1:98465987665415
        },
        {
            contact2:984561556415

        }

    ],
    join_date: ISODate('2022-06-14T12:30:00.000Z')
},
{
    staff_id        : "st6",
    staff_name      : "Kopila Magar",
    gender          : "Female",
    staff_role      : "Cleaning",
    staff_address   : "Birgunj",
    contact_number  : [
        {
            contact1:98461651
        },
        {
            contact2:98456545

        }

    ],
    join_date       : ISODate('2022-06-17T12:30:00.000Z')
},
{
```

```
        staff_id       : "st7",
        staff_name     : "Shiva Puri",
        gender         : "Male",
        staff_role     : "Lab Assistance",
        staff_address  : "Sagarmatha",
        contact_number: [
            {
                contact1:98466215448
            },
            {
                contact2:9845654118

            }

        ],
        join_date      : ISODate('2022-06-20T12:30:00.000Z')
    },
    {
        staff_id        : "st8",
        staff_name      : "Salin Thakuri",
        gender          : "Male",
        staff_role      : "Data Administrator",
        staff_addres    : "Kathmandu",
        contact_number :  [
            {
                contact1:984659448
            },
            {
                contact2:9845655215

            }

        ],
        join_date       : ISODate('2022-06-26T12:30:00.000Z')
    },
    {
        staff_id        : "st9",
        staff_name      : "Kabita Luitel",
        gender          : "Female",
        staff_role      : "Cleaning",
        staff_address   : "Pokhara",
        contact_number :  [
            {
                contact1:984659854588
            },
            {
                contact2:984565521551

            }

        ],
        join_date       : ISODate('2022-06-27T12:30:00.000Z')
    },
]);
```

Output:

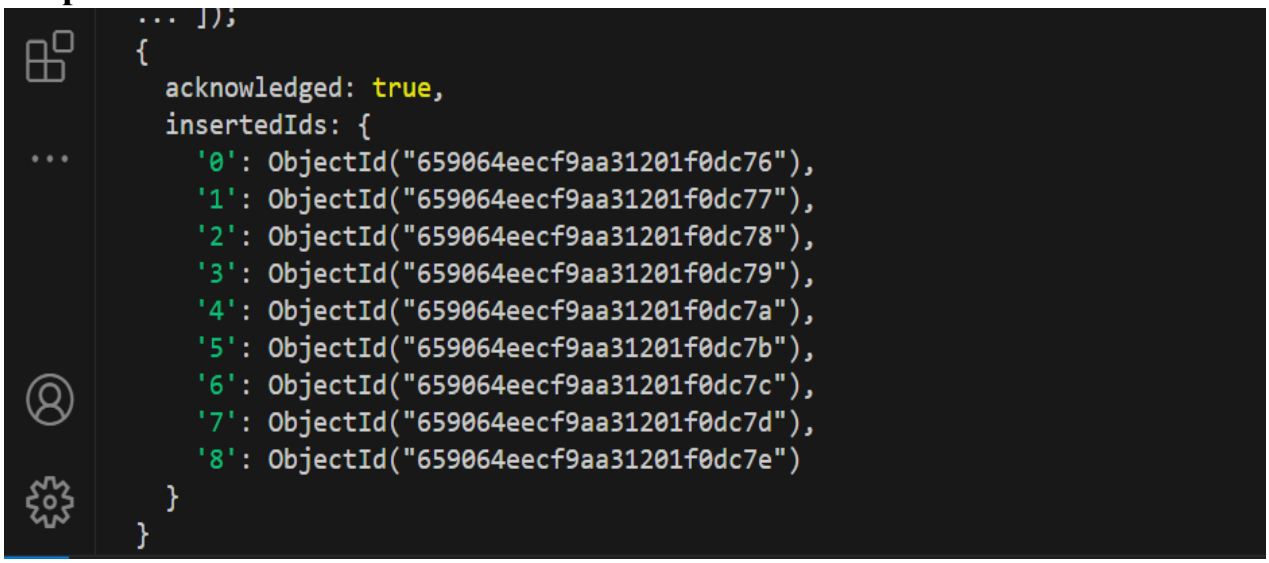

{
acknowledged: true,
insertedIds: {
'0': ObjectId("65906495cf9aa31201f0dc6d"),
'1': ObjectId("65906495cf9aa31201f0dc6e"),
'2': ObjectId("65906495cf9aa31201f0dc6f"),
'3': ObjectId("65906495cf9aa31201f0dc70"),
'4': ObjectId("65906495cf9aa31201f0dc71"),
'5': ObjectId("65906495cf9aa31201f0dc72"),
'6': ObjectId("65906495cf9aa31201f0dc73"),
'7': ObjectId("65906495cf9aa31201f0dc74"),
'8': ObjectId("65906495cf9aa31201f0dc75")
}
}
CollegeDataBase>

Query:

```
db.createCollection("Staff_Account");
db.Staff_Account.insertMany([
  {
      "STAFF_ACCOUNT_ID": "st5",
      "TOTAL_SALARY": 800,
      "REMAINING_SALARY": 650,
      "LAST_TRANSACTION_DATE": ISODate('2023-12-30T10:26:44.777Z'),
      "SALARY_TAKEN": 150
  },
  {
      "STAFF_ACCOUNT_ID": "st6",
      "TOTAL_SALARY": 900,
      "REMAINING_SALARY": 740,
      "LAST_TRANSACTION_DATE": ISODate('2023-12-30T10:26:44.784Z'),
      "SALARY_TAKEN": 160
  },
  {
      "STAFF_ACCOUNT_ID": "st7",
      "TOTAL_SALARY": 1000,
      "REMAINING_SALARY": 950,
      "LAST_TRANSACTION_DATE": ISODate('2023-12-30T10:26:44.791Z'),
      "SALARY_TAKEN": 50
  },
  {
      "STAFF_ACCOUNT_ID": "st8",
      "TOTAL_SALARY": 700,
      "REMAINING_SALARY": 0,
      "LAST_TRANSACTION_DATE": ISODate('2023-12-30T10:26:44.795Z'),
      "SALARY_TAKEN": 700
  },
  {
      "STAFF_ACCOUNT_ID": "st9",
      "TOTAL_SALARY": 700,
      "REMAINING_SALARY": 700,
      "LAST_TRANSACTION_DATE": null,
      "SALARY_TAKEN": 0
  },
  {
      "STAFF_ACCOUNT_ID": "st1",
      "TOTAL_SALARY": 800,
      "REMAINING_SALARY": 600,
      "LAST_TRANSACTION_DATE": ISODate('2023-12-30T10:26:44.754Z'),
      "SALARY_TAKEN": 200
  },
  {
      "STAFF_ACCOUNT_ID": "st2",
      "TOTAL_SALARY": 900,
      "REMAINING_SALARY": 300,
      "LAST_TRANSACTION_DATE": ISODate('2023-12-30T10:26:44.761Z'),
      "SALARY_TAKEN": 600
  },
  {
```

```
        "STAFF_ACCOUNT_ID": "st3",
        "TOTAL_SALARY": 1000,
        "REMAINING_SALARY": 500,
        "LAST_TRANSACTION_DATE": ISODate('2023-12-30T10:26:44.766Z'),
        "SALARY_TAKEN": 500
    },
    {
        "STAFF_ACCOUNT_ID": "st4",
        "TOTAL_SALARY": 700,
        "REMAINING_SALARY": 400,
        "LAST_TRANSACTION_DATE": ISODate('2023-12-30T10:26:44.773Z'),
        "SALARY_TAKEN": 300
    }
]);
```

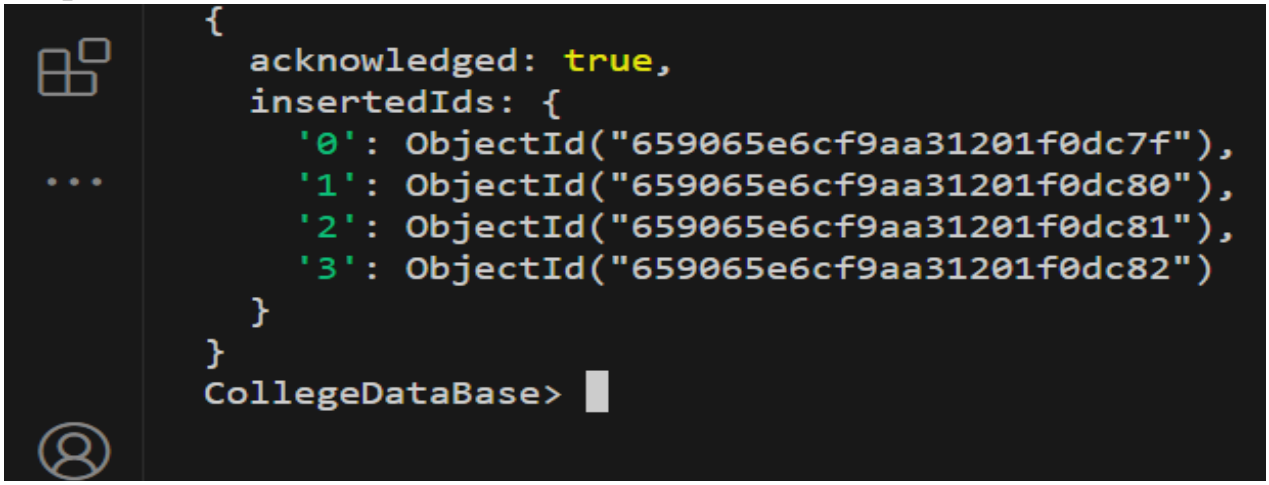
Output:



Query:

```
db.createCollection("Sport_Items");
db.Sport_Items.insertMany([
    { Item_Name: 'Football',
     Quantity: 5
    },
    { Item_Name: 'Basketball',
     Quantity: 5
    },
    { Item_Name: 'TT',
    Quantity: 20
    },
    { Item_Name: 'CricketSet',
     Quantity: 2
    }
  ]);
```

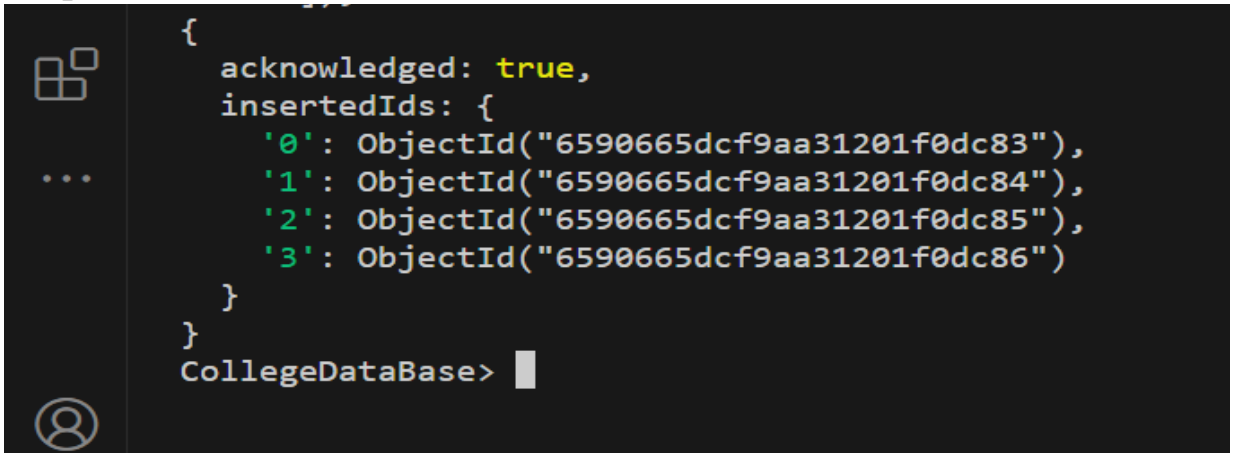
Output:



Query:

```
db.createCollection("Books_Items");
db.Books_Items.insertMany([
  { Item_Name: 'C++',
   Quantity: 50
   },
  { Item_Name: 'DSA',
   Quantity: 50
  },
  { Item_Name: 'Python',
   Quantity:50
  },
  { Item_Name: 'Calculas',
   Quantity: 50
   }
]);
```

Output:



Query:

```
db.createCollection("Sport_Items__Issued_and_Return_Item")
db.Sport_Items__Issued_and_Return_Item.insertMany([
    {
        "ISSUDER_ID": "s1",
        "ISSED_ITEM": "Football",
        "ISSED_QUANTITY": 3,
        "ISSUED_DATE": "2022-09-01T12:00:00.000Z",
        "RETURN_DATE": "2022-10-01T12:00:00.000Z",
        "DAYS_ISSUED": 30,
        "RETURN_STATTUS": "YES",
        "FINE_AMOUNT": 100
      },
      {
        "ISSUDER_ID": "s2",
        "ISSED_ITEM": "TT",
        "ISSED_QUANTITY": 3,
        "ISSUED_DATE": "2022-09-01T12:00:00.000Z",
        "RETURN_DATE": null,
        "DAYS_ISSUED": null,
        "RETURN_STATTUS": null,
        "FINE_AMOUNT": null
      }

])
```

Output:

```
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("659066ffcf9aa31201f0dc87"),
    '1': ObjectId("659066ffcf9aa31201f0dc88")
  }
}
CollegeDataBase>
```

Descriptions:

The NOSQL code provided operates in the MongoDB environment, primarily focusing on the "CollegeDataBase." It begins by dropping existing collections related to college information, students, teachers, staff, sport items, books, and their corresponding accounts to create a fresh database. Subsequently, collections are created and populated with sample data for college information, students, teachers, staff, sport items, and books.

Several aggregation queries are then performed. The first one involves joining student information with student accounts and sport items issued and returned. It filters students based on remaining fees and fine amounts. The second query uses the union operation to combine results from the teacher and staff collections, filtering based on remaining salary. The third query employs a nested query to find students with fine amounts in the sport items collection.

Additionally, there's a date-based query to find teachers who joined within the last year, and an OLAP feature cube aggregation for staff salaries based on gender. The provided code serves as a comprehensive example of MongoDB operations, including data manipulation and aggregation.

Comparison of Scripts

The comparison between Oracle SQL query and MongoDB is given below:

TASK FOUR: JOIN

Query 1 -A join of three or More tables With restrictions.
The first query is a SQL join query that selects columns from three tables: Student, Student_Account,Sport_Items__Issued_and_Return_Item.It uses left join and right join operators to join the tables and a where clause to filter results.

The Second query is a MongoDB aggregation query that uses $lookup operators to perform joins between the Student,Student_Account and Sport_Items__Issued_and_Return_Item.It includes a $match operator to filter results and a $project operator to select fields.
The goal of both queries is to retrieve information about Student, Student_Account and Sport_Items__Issued_and_Return_Item whose remaining_fee >3000 and Fine_Amount is greater than 50

SQL Query	MongoDB Script

select
Student.student_name,
Student.student_department,
Student.student_current_semester,
Student.student_address,
Student.student_contact_number,
Student_Account.remaining_fee,
 	
Sport_Items__Issued_and_Return_Item.Issed_item,
 	
Sport_Items__Issued_and_Return_Item.Issed_quantity,
 	
Sport_Items__Issued_and_Return_Item.Fine_Amount
From Student
Left join Student_Account on
Student.Student_id =
Student_Account.student_account_id
	db.Student.aggregate([
	{
	$lookup:{
	from:'Student_Account',
	localField:'student_id',
	
	foreignField:'STUDENT_ACCOUNT_ID',
	as:'stu'
	}
	},
	{
	$lookup:{
	
	from:'Sport_Items__Issued_and_Return_Item',
	localField:'student_id',
	foreignField:'ISSUDER_ID',
	as:'stusp'
	}
	},

Right join
Sport_Items__Issued_and_Return_Item on
Student.Student_id =
Sport_Items__Issued_and_Return_Item.Issuder_id
where Student_Account.remaining_fee >3000 and
Sport_Items__Issued_and_Return_Item.Fine_Amount >50;
	{
	$match:{
	'stu.REMAINING_FEE' :{$gt:3000},
	'stusp.FINE_AMOUNT' :{$gt:50}
	}
	},
	{
	$project:{
	"student_id":1,
	"student_name":1,
	"student_department":1,
	"student_current_semester":1,
	"student_address":1,
	"student_contact_number":1,
	"stu.REMAINING_FEE":1,
	"stusp.ISSED_ITEM": 1,
	"stusp.ISSED_QUANTITY": 1,
	"stusp.FINE_AMOUNT": 1
	}
	}
	
	]);

SQL output:

STUDENT_NAME	STUDENT_DEPARTMENT	STUDENT_CURRENT_SEMESTER	STUDENT_ADDRESS	STUDENT_CONTACT_NUMBER	REMAINING_FEE	ISSED_ITEM	ISSED_QUANTITY	FINE_AMOUNT
Ram Poudel	IT	3rd	Kathmandu	9845697859	5000	Football	3	100

Download CSV

Mongodb Output:

```
  {
    _id: ObjectId("65906171cf9aa31201f0dc5f"),
    student_id: 's1',
    student_name: 'Ram Poudel',
    student_department: 'IT',
    student_current_semester: '3rd',
    student_address: 'Kathmandu',
    student_contact_number: [ { contact: 9845697859 }, { contact: 9846456456 } ],
    stu: [ { REMAINING_FEE: 5000 } ],
    stusp: [ { ISSED_ITEM: 'Football', ISSED_QUANTITY: 3, FINE_AMOUNT: 100 } ]
  }
]
CollegeDataBase>
```

TASK FIVE : UNION

Query 2: A query which uses one (or more) of UNION,DIFFERENCE or INTERSECTION

Both queries retrieve the data of Teacher,Teacher_account and Staff,Staff_account details whose remaining salary is greater than 700$ and 500$ respectively.

The first query is a SQL Union query that selects all columns from the tables Teacher and Teacher_account Union with Tables Staff and Staff_account joining them . It uses the UNION keyword to combine the results of two separate queries.

The second query is a Mongodb aggregation operation with Teacher and Teacher_Account collection along with $unionwith operator is used to join the Staff and Staff_Account collections $match operator is used to filter.Likewise $project operator is used for selection of fields.

SQL query	MongoDB Script

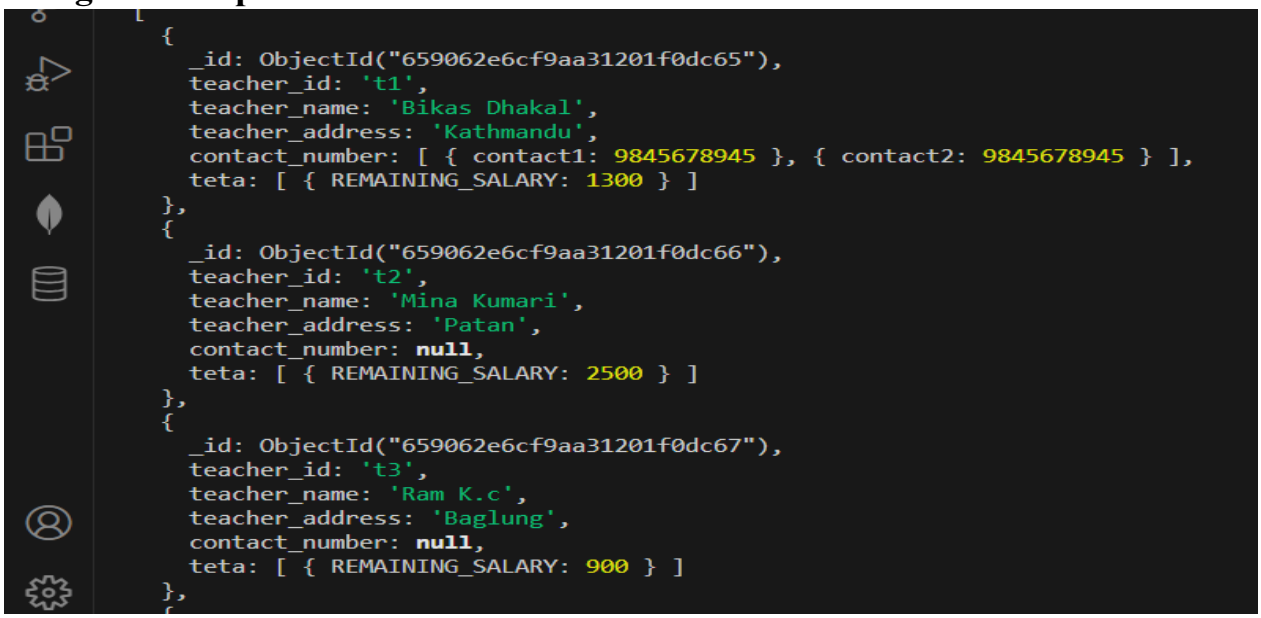
select
Teacher.teacher_id,
Teacher.teacher_name,
Teacher.teacher_address,
Teacher.contact_number,
Teacher_account.remaining_salary
From Teacher
Left join Teacher_account on
Teacher.teacher_id =
Teacher_account.teacher_account_id
where remaining_salary >700
Union
Select
Staff.staff_id,
Staff.staff_name,
Staff.staff_address,
Staff.contact_number,
Staff_account.remaining_salary
From Staff
Right Join Staff_account on Staff.staff_id =
Staff_account.staff_account_id
where Staff_account.remaining_salary >
500;
	db.Teacher.aggregate([
	{
	$lookup:{
	from:"Teacher_Account",
	localField:"teacher_id",
	foreignField:"TEACHER_ACCOUNT_ID",
	as:"teta"
	}
	},
	{
	$match:{
	"teta.REMAINING_SALARY"
	:{$gt:700}
	}
	},
	{
	$project:{
	"teacher_id":1,
	"teacher_name":1,
	"teacher_address":1,
	"contact_number":1,
	"teta.REMAINING_SALARY":1
	}
	},
	{
	$unionWith:{
	coll:"Staff",
	pipeline:[
	{
	$lookup:{
	from:"Staff_Account",
	localField:"staff_id",
	foreignField:"STAFF_ACCOUNT_ID",
	as:"stac"

```
            }
        },
        {
            $match:{

"stac.REMAINING_SALARY":{$gt:500}

            }
        },

        {
            $project:{
                "staff_id":1,
                "staff_name":1,
                "staff_address":1,
                "contact_number":1,

"stac.REMAINING_SALARY":1



            }
        }
    ]
  }
 }
])
```

SQL Output:

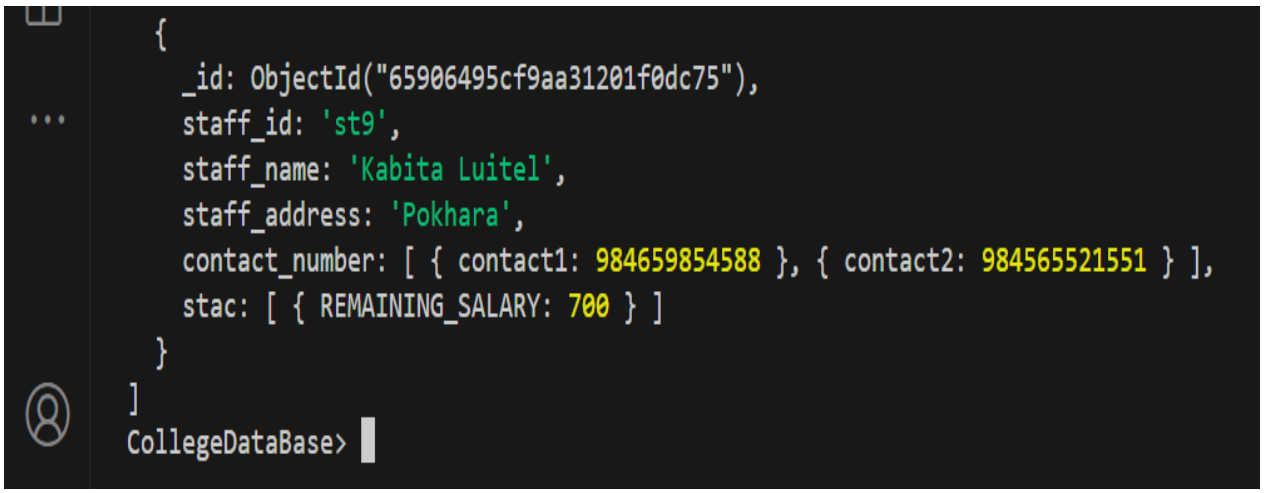
TEACHER_ID	TEACHER_NAME	TEACHER_ADDRESS	CONTACT_NUMBER	REMAINING_SALARY
st1	Shuba Tamang	Kathmandu	98465987623	600
st5	Ravi Magar	Pokhara	98465987665415	650
st6	Kopila Magar	Birgunj	98461651	740
st7	Shiva Puri	Sagarmatha	98466215448	950
st9	Kabita Luitel	Pokhara	984659854588	700
t1	Bikas Dhakal	Kathmandu	9845678945	1300
t2	Mina Kumari	Patan	9845678874	2500
t3	Ram K.c	Baglung	9845678569	900

MongoDB Output:



```
[
  {
    _id: ObjectId("659062e6cf9aa31201f0dc65"),
    teacher_id: 't1',
    teacher_name: 'Bikas Dhakal',
    teacher_address: 'Kathmandu',
    contact_number: [ { contact1: 9845678945 }, { contact2: 9845678945 } ],
    teta: [ { REMAINING_SALARY: 1300 } ]
  },
  {
    _id: ObjectId("659062e6cf9aa31201f0dc66"),
    teacher_id: 't2',
    teacher_name: 'Mina Kumari',
    teacher_address: 'Patan',
    contact_number: null,
    teta: [ { REMAINING_SALARY: 2500 } ]
  },
  {
    _id: ObjectId("659062e6cf9aa31201f0dc67"),
    teacher_id: 't3',
    teacher_name: 'Ram K.c',
    teacher_address: 'Baglung',
    contact_number: null,
    teta: [ { REMAINING_SALARY: 900 } ]
  },
```

```
  _id: ObjectId("65906495cf9aa31201f0dc6d"),
  staff_id: 'st1',
  staff_name: 'Shuba Tamang',
  staff_address: 'Kathmandu',
  contact_number: [ { contact1: 98465987623 }, { contact2: 98456151454 } ],
  stac: [ { REMAINING_SALARY: 600 } ]
},
{
  _id: ObjectId("65906495cf9aa31201f0dc71"),
  staff_id: 'st5',
  staff_name: 'Ravi Magar',
  staff_address: 'Pokhara',
  contact_number: [ { contact1: 98465987665415 }, { contact2: 984561556415 } ],
  stac: [ { REMAINING_SALARY: 650 } ]
},
{
  _id: ObjectId("65906495cf9aa31201f0dc72"),
  staff_id: 'st6',
  staff_name: 'Kopila Magar',
  staff_address: 'Birgunj',
  contact_number: [ { contact1: 98461651 }, { contact2: 98456545 } ],
  stac: [ { REMAINING_SALARY: 740 } ]
},
{
  _id: ObjectId("65906495cf9aa31201f0dc73"),
  staff_id: 'st7',
  staff_name: 'Shiva Puri',
  staff_address: 'Sagarmatha',
  contact_number: [ { contact1: 98466215448 }, { contact2: 9845654118 } ],
  stac: [ { REMAINING_SALARY: 950 } ]
```



TASK SIX : NESTED QUERIES

Query 3 : A query which requires use of either a nested table or subtypes .

The First query is SQL Query which select student_id,student_name, student_department, Student_current_semester,student_address,student_contact_number from Student Table by matching student_id of this table to the Sport_Items__Issued_and_Return_Item.Issuder_id column whose Fine_Amount is greater than 50$.

The second query is Mongodb script in which two collections Student and Sport_Items__Issued_and_Return_Itemcollection are join using $lookup ,to filter $match is used and to select the fields $project is used.

The goal of two queries is to extract the Member information which is absent in one table or collection by making the next table or collection nested .

SQL Query

```
select
Student.student_id,Student.student_name,Stu
dent.student_department,
Student.student_current_semester,Student.stu
dent_address,Student.student_contact_number
from Student
Where Student.student_id In(
   Select
Sport_Items__Issued_and_Return_Item.Issud
er_id
   from
Sport_Items__Issued_and_Return_Item
   where
Sport_Items__Issued_and_Return_Item.Fine_
Amount>=50
);
```

MongoDb Script

```
db.Student.aggregate([
   {
      $lookup:{

from:"Sport_Items__Issued_and_Return_Item
",
         localField:"student_id",
         foreignField:"ISSUDER_ID",
         as:"stusport"
      }
   },
   {
      $match:{
         "stusport.FINE_AMOUNT":{$gte:50}
      }
   },
   {
      $project:{
         student_id:1,
         student_name:1,
         student_department:1,
         student_current_semester:1,
         student_address:1,
         student_contact_number:1


      }
   }
])
```

SQL Output:

STUDENT_ID	STUDENT_NAME	STUDENT_DEPARTMENT	STUDENT_CURRENT_SEMESTER	STUDENT_ADDRESS	STUDENT_CONTACT_NUMBER
s1	Ram Poudel	IT	3rd	Kathmandu	9845697859

Download CSV

MongoDB Output:

TASK SEVEN : TEMPORAL FEATURE QUERY

Query 4 : A query using temporal features (e.g., timestamps, intervals, etc.) of Oracle SQL

The first query is SQL which uses the temporal feature Current_Timestamp.In this section this feature is used to return all teachers that were experienced of less than one year in college.

The second script is Mongodb.In this script $expr operator is used for aggregation expressions ,$and operator contains two conditions:
$lte to check if 'joined_date' is less than or equal to one year ago from current date,
$lte to check if 'joined date' is less than the current date
Likewise, Projection is used for selecting fields.
The goal of two queries is to find the Teachers who joined and have not more than one year.

SQL Query	MongoDB Script

SELECT
teacher_id,
teacher_name,
teacher_address,
contact_number
From
Teacher
where
join_date <= CURRENT_TIMESTAMP - INTERVAL '1' Year;
	var query ={
	$expr:{
	$and:[
	{
	$lte :["$join_date",{$subtract:[new Date(),{$multiply:[365,24,60,60,1000]}]}]
	},
	{
	$lt :["$join_date",new Date()]
	}
	]
	}
	};
	
	var projection ={
	teacher_id:1,
	teacher_name:1,
	teacher_address:1,
	contact_number:1
	
	}
	var result = db.Teacher.find(query,projection);
	result.forEach(function (doc) {
	printjson(doc);
	});

SQL Output:

TEACHER_ID	TEACHER_NAME	TEACHER_ADDRESS	CONTACT_NUMBER
t1	Bikas Dhakal	Kathmandu	9845678945
t2	Mina Kumari	Patan	9845678874
t3	Ram K.c	Baglung	9845678569
t4	Sita Pradhan	Kathmandu	9845678666

Download CSV

MongoDB Output:

```
    _id: ObjectId("659062e6cf9aa31201f0dc65"),
    teacher_id: 't1',
    teacher_name: 'Bikas Dhakal',
    teacher_address: 'Kathmandu',
    contact_number: [ { contact1: 9845678945 }, { contact2: 9845678945 } ]
  }
  {
    _id: ObjectId("659062e6cf9aa31201f0dc66"),
    teacher_id: 't2',
    teacher_name: 'Mina Kumari',
    teacher_address: 'Patan',
    contact_number: null
  }
  {
    _id: ObjectId("659062e6cf9aa31201f0dc67"),
    teacher_id: 't3',
    teacher_name: 'Ram K.c',
    teacher_address: 'Baglung',
    contact_number: null
  }
  {
    _id: ObjectId("659062e6cf9aa31201f0dc68"),
    teacher_id: 't4',
    teacher_name: 'Sita Pradhan',
    teacher_address: 'Kathmandu',
    contact_number: null
  }

CollegeDataBase>
```

TASK EIGHT : TEMPORAL FEATURE QUERY

Query 5: A query using OLAP (e.g., ROLLUP, CUBE, PARTITION) features of Oracle SQL

The first Query is SQL which uses the Partition OLAP features.Cube feature is used to find Total_Salary using Group by the male and female gender and also find running total.

The second Query is a NOSQL query. Aggregate function is used along with $group to group by gender ,likewise $unwind,$project feature is used in a Staff_Account collection.

The goal of both queries is to find total salaries based upon gender

SQL Query	MongoDB Script

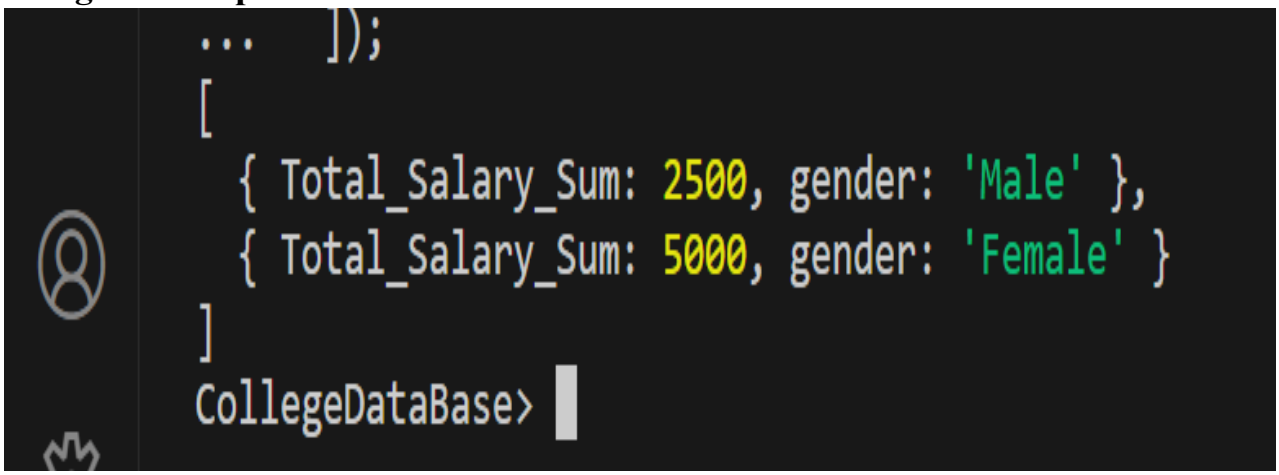
SELECT
gender,
sum(TOTAL_SALARY) as Total_Salary_sum
from Staff s
join
Staff_account sa on s.Staff_id =
sa.STAFF_ACCOUNT_ID
group by
cube(Gender);
	db.Staff.aggregate([
	{
	$lookup: {
	from: "Staff_Account",
	localField: "staff_id",
	foreignField: "STAFF_ACCOUNT_ID",
	as: "staffAccount"
	}
	},
	{
	$unwind: "$staffAccount"
	},
	{
	$group: {
	_id: "$gender",
	Total_Salary_Sum: { $sum:
	"$staffAccount.TOTAL_SALARY" }
	}
	},
	{
	$project: {
	gender: "$_id",
	Total_Salary_Sum: 1,
	_id: 0
	}
	}
	]);

SQL output:

GENDER	TOTAL_SALARY_SUM
-	7500
Male	2500
Female	5000

Download CSV

MongoDB Output:



```
...   ]);
[
  { Total_Salary_Sum: 2500, gender: 'Male' },
  { Total_Salary_Sum: 5000, gender: 'Female' }
]
CollegeDataBase>
```

Conclusion:

Conclusion:

In conclusion, I have successfully created and run both an Oracle SQL and MongoDB database queries , and implement five queries on each to evaluate their performance.I performed queries such as joins,union,nested query,To_Timestamp, and OLAP(cube)and made sure that the result produced by both database result were similar.

After evaluating the performance of both databases,I found that they were efficient and capable of handling the queries with ease.However, the choice depends upon the needs and requirements of the project.

If a project requires structured data and adherence to ACID properties for robust transactional support, SQL databases are the preferred choice. On the contrary, in scenarios where a flexible schema and rapid development are paramount, NoSQL databases prove to be the most suitable option.

REFERENCES:

Author(s): Oracle Corporation
Year of publication: 2001 (you can use the last update date if available)
Title of the document: Oracle® Database PL/SQL User's Guide and Reference
URL: https://docs.oracle.com/cd/B10500_01/appdev.920/a96624/toc.htm

MongoDB Documentation
URL: https://www.mongodb.com/docs/

Boicea, A., Radulescu, F. and Agapin, L.I. (2012) "MongoDB vs oracle -- database comparison," 2012 Third International Conference on Emerging Intelligent Data and Web Technologies [Preprint]. Available at:
https://doi.org/10.1109/eidwt.2012.32.