

NTechnological Institute of the Philippines
938 Aurora Blvd. Cubao, Quezon City

College of Computer Studies

ITE 014 – Information Management

Final Period

Group Number: Name: Faurillo, Ymnwl Jan P. Tiamzon, Bryan Dominick Ureta, Juster	Date: 05/10/2023
Program / Section: ITE 014 – Information Management CS22S2	Instructor: Ms. Roxanne A. Pagaduan
Assessment Task: Project Part 4: Creating Tables and Managing Data	

Requirements:

1. Collaborate with your team members and satisfy the following:
 - Create tables through SQL statements.
 - Provide the table structures (using describe function) of your database.
 - Insert records. (provide at least 30 transactions)
 - Add Constraints as applicable.
 - Note: Apply best practices in creating, inserting, updating, and managing data in your database.
2. Provide the specific contributions of each member. (use the template provided)
3. Submit all related files including .sql files used in the project.
4. Save your file as TeamName_ProjectPart3.DOCX and .PDF.

ANSWER:

TASK 1: Table Structure

Table Name 1: Employees

The screenshot shows the Oracle Live SQL interface. On the left, there is a sidebar with navigation options: Home, SQL Worksheet (selected), My Session, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The main area displays the 'DESC employees' command and the resulting table structure. The table has 12 columns: employee_id, first_name, middle_name, last_name, birth_date, salary, commission_pct, phone_number, email, hire_date, and department_id. Each column's data type and nullability are listed.

Column	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(3,0)
FIRST_NAME	NOT NULL	VARCHAR2(30)
MIDDLE_NAME	NOT NULL	VARCHAR2(30)
LAST_NAME	NOT NULL	VARCHAR2(30)
BIRTH_DATE	NOT NULL	DATE
SALARY	NOT NULL	NUMBER(8,2)
COMMISSION_PCT		NUMBER(5,2)
PHONE_NUMBER	NOT NULL	NUMBER(11,0)
EMAIL	NOT NULL	VARCHAR2(50)
HIRE_DATE	NOT NULL	DATE
DEPARTMENT_ID	NOT NULL	NUMBER(3,0)

Download CSV

2023 Oracle Live SQL 23.1.5, running Oracle Database 19c EE Software Patch 19.07.0000 Database Documentation Ask Tom Dev Open

This SQL query creates a table named "Employees" with the following columns:

- employee_id: a non-null integer column with a maximum value of 3 digits, which is the primary key for the table.
- first_name: a non-null string column with a maximum length of 30 characters, representing the employee's first name.
- middle_name: a non-null string column with a maximum length of 30 characters, representing the employee's middle name.
- last_name: a non-null string column with a maximum length of 30 characters, representing the employee's last name.
- birth_date: a non-null date column representing the employee's date of birth.
- salary: a non-null numeric column with a maximum value of 8 digits, 2 of which are decimals, representing the employee's salary.
- commission_pct: a numeric column with a maximum value of 5 digits, 2 of which are decimals, representing the employee's commission percentage. This column can be null.
- phone_number: a non-null numeric column with a maximum value of 11 digits, representing the employee's phone number. This column also has a unique constraint to ensure that no two employees have the same phone number.
- email: a non-null string column with a maximum length of 50 characters, representing the employee's email address. This column also has a unique constraint to ensure that no two employees have the same email address.
- hire_date: a non-null date column representing the employee's hire date.
- department_id: a non-null integer column with a maximum value of 3 digits, representing the department that the employee belongs to.

The table creation statement also includes some constraints on certain columns to ensure data integrity. The primary key constraint on the "employee_id" column ensures that each employee has a unique identifier. The unique constraint on the "phone_number" and "email" columns ensures that no two employees have the same phone number or email address.

Table Name 2: Loan_Allocations

The screenshot shows a web-based SQL editor interface. At the top, there's a header with 'Live SQL' and navigation links like 'Home', 'SQL Worksheet', 'My Session', 'Schema', 'Quick SQL', 'My Scripts', 'My Tutorials', and 'Code Library'. The main area displays a SQL query: `CREATE TABLE loan_allocations`. Below the query, a table structure is shown with columns: `loan_id` (NUMBER(7,0)), `loan_category` (VARCHAR2(30)), `loan_startpayment` (DATE), `loan_endpayment` (DATE), `loan_amount` (NUMBER(8,2)), `interest_rate` (NUMBER(2,2)), and `down_payment` (NUMBER(8,2)). A 'Download CSV' button and '7 rows selected' message are also visible. The bottom status bar indicates the database is Oracle Live SQL 23.1.5.

column	nullable	type
loan_id	NOT NULL	NUMBER(7,0)
loan_category	NOT NULL	VARCHAR2(30)
loan_startpayment	NOT NULL	DATE
loan_endpayment	NOT NULL	DATE
loan_amount	NOT NULL	NUMBER(8,2)
interest_rate	NOT NULL	NUMBER(2,2)
down_payment	NOT NULL	NUMBER(8,2)

This SQL query creates a table named "Loan_Allocations" with the following columns:

- `loan_id`: a non-null integer column with a maximum value of 7 digits, which is the primary key for the table.
- `loan_category`: a non-null string column with a maximum length of 30 characters, representing the category of the loan.
- `loan_startpayment`: a non-null date column representing the start date of the loan payments.
- `loan_endpayment`: a non-null date column representing the end date of the loan payments.
- `loan_amount`: a non-null numeric column with a maximum value of 8 digits, 2 of which are decimals, representing the total amount of the loan.
- `interest_rate`: a non-null numeric column with a maximum value of 2 digits, 2 of which are decimals, representing the interest rate of the loan.
- `down_payment`: a non-null numeric column with a maximum value of 8 digits, 2 of which are decimals, representing the amount of down payment required for the loan.
- The table creation statement also includes a primary key constraint on the "loan_id" column to ensure that each loan allocation has a unique identifier.

This table seems to represent a loan allocation system where loans are categorized, have specific start and end dates for payments, and are associated with specific interest rates and down payments.

Table Name 3: Borrowers

The screenshot shows the 'Live SQL' web application interface. On the left, there is a sidebar with navigation links: Home, SQL Worksheet, My Session, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The main area is titled 'SQL Worksheet' and contains a single SQL statement: `CREATE TABLE borrowers`. Below the SQL editor, a table schema for 'TABLE BORROWERS' is displayed. The table has 12 columns: borrower_id, first_name, middle_name, last_name, birth_date, address, phone_number, monthly_salary, loan_id, employee_id, and payment_id. Each column's data type, nullability, and constraints are listed. At the bottom of the interface, a Windows taskbar is visible with various application icons and a system clock showing 3:22 AM on 10/09/2023.

Column	Null?	Type
borrower_id	NOT NULL	NUMBER(7,0)
first_name	NOT NULL	VARCHAR2(30)
middle_name	NOT NULL	VARCHAR2(30)
last_name	NOT NULL	VARCHAR2(30)
birth_date	NOT NULL	DATE
address	NOT NULL	VARCHAR2(70)
phone_number	NOT NULL	NUMBER(11,0)
monthly_salary	NOT NULL	NUMBER(9,0)
loan_id	NOT NULL	NUMBER(7,0)
employee_id	NOT NULL	NUMBER(3,0)
payment_id	NOT NULL	NUMBER(7,0)

This SQL query creates a table named "Borrowers" with the following columns:

- **borrower_id:** a non-null integer column with a maximum value of 7 digits, which is the primary key for the table.
- **first_name:** a non-null string column with a maximum length of 30 characters, representing the borrower's first name.
- **middle_name:** a non-null string column with a maximum length of 30 characters, representing the borrower's middle name.
- **last_name:** a non-null string column with a maximum length of 30 characters, representing the borrower's last name.
- **birth_date:** a non-null date column representing the borrower's date of birth.
- **address:** a non-null string column with a maximum length of 70 characters, representing the borrower's address.
- **phone_number:** a non-null numeric column with a maximum value of 11 digits, representing the borrower's phone number. This column also has a unique constraint to ensure that no two borrowers have the same phone number.
- **monthly_salary:** a non-null numeric column with a maximum value of 9 digits, representing the borrower's monthly salary.
- **loan_id:** a non-null integer column with a maximum value of 7 digits, representing the loan allocation ID that the borrower is associated with. This column also has a foreign key constraint referencing the "loan_id" column in the "Loan_Allocations" table.
- **employee_id:** a non-null integer column with a maximum value of 3 digits, representing the employee ID of the employee who helped the borrower obtain the loan. This column also has a foreign key constraint referencing the "employee_id" column in the "Employees" table.
- **payment_id:** a non-null integer column with a maximum value of 7 digits, representing the payment ID associated with the borrower's loan payments.

The table creation statement also includes some constraints on certain columns to ensure data integrity. The primary key constraint on the "borrower_id" column ensures that each borrower has a unique identifier. The unique constraint on the "phone_number" column ensures that no two borrowers have the same phone number. Additionally, the "loan_id" and "employee_id" columns have foreign key constraints to ensure that the values inserted in those columns reference existing values in the "Loan_Allocations" and "Employees" tables, respectively.

Table Name 4: Payments

The screenshot shows a web-based SQL Worksheet interface. On the left, there is a sidebar with navigation links: Home, SQL Worksheet, My Session, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The main area displays the 'payments' table structure. Below the table structure, there is a 'Download CSV' button and a message '7 rows selected.' At the bottom of the interface, there is a status bar with version information and a timestamp.

Column	Null?	Type
PAYMENT_ID	NOT NULL	NUMBER(7,0)
LOAN_ID	NOT NULL	NUMBER(7,0)
BORROWER_ID	NOT NULL	NUMBER(7,0)
REFERENCE_ID	NOT NULL	NUMBER(7,0)
PAYMENT_AMOUNT	NOT NULL	NUMBER(8,2)
PAYMENT_METHOD	NOT NULL	VARCHAR2(30)
PAYMENT_STATUS	NOT NULL	VARCHAR2(30)

Download CSV
7 rows selected.

2023 Oracle - Live SQL 23.1.1, running Oracle Database 19c EE Extreme Perf - 19.07.000 - Database Documentation - Ask Tom - Dev Open
Run with using Oracle JDBC - Please - Terms of Use

Watchlist Ideas 3:27 AM 10/09/2023

This SQL query creates a table named "Payments" with the following columns:

- payment_id: a non-null integer column with a maximum value of 7 digits, which is the primary key for the table.
- loan_id: a non-null integer column with a maximum value of 7 digits, representing the loan allocation ID associated with the payment. This column also has a foreign key constraint referencing the "loan_id" column in the "Loan_Allocations" table.
- borrower_id: a non-null integer column with a maximum value of 7 digits, representing the borrower ID associated with the payment. This column also has a foreign key constraint referencing the "borrower_id" column in the "Borrowers" table.
- reference_id: a non-null integer column with a maximum value of 7 digits, representing the unique reference ID associated with the payment.
- payment_amount: a non-null numeric column with a maximum value of 8 digits and 2 decimal places, representing the payment amount.
- payment_method: a non-null string column with a maximum length of 30 characters, representing the payment method used to make the payment.
- payment_status: a non-null string column with a maximum length of 30 characters,

representing the current status of the payment.

The table creation statement also includes some constraints on certain columns to ensure data integrity. The primary key constraint on the "payment_id" column ensures that each payment has a unique identifier. The foreign key constraints on the "loan_id" and "borrower_id" columns ensure that the values inserted in those columns reference existing values in the "Loan_Allocations" and "Borrowers" tables, respectively.

Table Name 5: Payments_History

The screenshot shows a web-based SQL Worksheet interface. On the left, there is a sidebar with navigation links: Home, SQL Worksheet, My Session, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The main area displays the table structure for 'payments_history'. Below the table structure, there is a 'Download CSV' button and a note '7 rows selected.' At the bottom of the interface, there is a status bar indicating the database version and other details.

Column	Null?	Type
REFERENCE_ID	NOT NULL	NUMBER(7,0)
PAYMENT_ID	NOT NULL	NUMBER(7,0)
LOAN_ID	NOT NULL	NUMBER(7,0)
BORROWER_ID	NOT NULL	NUMBER(7,0)
LATE_FEES	-	NUMBER(8,2)
BALANCE_BEFORE_PAYMENT	-	NUMBER(8,2)
BALANCE_AFTER_PAYMENT	-	NUMBER(8,2)

This SQL query creates a table named "Payments_History" with the following columns:

- **reference_id:** a non-null integer column with a maximum value of 7 digits, which is the primary key for the table and represents the unique reference ID associated with a payment.
- **payment_id:** a non-null integer column with a maximum value of 7 digits, representing the payment ID associated with the payment. This column also has a foreign key constraint referencing the "payment_id" column in the "Payments" table.
- **loan_id:** a non-null integer column with a maximum value of 7 digits, representing the loan allocation ID associated with the payment. This column also has a foreign key constraint referencing the "loan_id" column in the "Loan_Allocations" table.
- **borrower_id:** a non-null integer column with a maximum value of 7 digits, representing the borrower ID associated with the payment. This column also has a foreign key constraint referencing the "borrower_id" column in the "Borrowers" table.
- **late_fees:** a numeric column with a maximum value of 8 digits and 2 decimal places, representing any late fees associated with the payment.
- **balance_before_payment:** a numeric column with a maximum value of 8 digits and 2 decimal places, representing the balance of the loan before the payment was made.
- **balance_after_payment:** a numeric column with a maximum value of 8 digits and 2 decimal places, representing the balance of the loan after the payment was made.

The table creation statement includes some constraints on certain columns to ensure data integrity. The primary key constraint on the "reference_id" column ensures that each payment has a unique reference ID. The foreign key constraints on the "payment_id", "loan_id", and "borrower_id" columns ensure that the values inserted in those columns reference existing values in the "Payments", "Loan_Allocations", and "Borrowers" tables, respectively.

TASK 2: Creating Tables (SQL Statements)

Table Name 1: Employees Table (Table Created)

The screenshot displays the Live SQL web application interface. On the left, a sidebar contains navigation links: Home, SQL Worksheet (selected), My Session, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The main area is titled 'SQL Worksheet' and contains the following SQL code:

```
1. CREATE TABLE Employees
2. (employee_id NUMBER (3) NOT NULL
3. CONSTRAINT employee_id_pk PRIMARY KEY,
4. first_name VARCHAR (30) NOT NULL,
5. middle_name VARCHAR (30) NOT NULL,
6. last_name VARCHAR (30) NOT NULL,
7. birth_date DATE NOT NULL,
8. salary NUMBER (8,2) NOT NULL,
9. commission_pct NUMBER (5,2),
10. phone_number NUMBER (11) NOT NULL,
11. CONSTRAINT phone_number_uk UNIQUE (phone_number),
12. email VARCHAR (50) NOT NULL,
13. CONSTRAINT email_uk UNIQUE (email),
14. hire_date DATE NOT NULL,
15. department_id NUMBER (3) NOT NULL
16. );
```

Below the code editor, a message states 'Table created.' with a green checkmark icon. At the bottom of the interface, a status bar indicates: '2023 Oracle - Live SQL 23.1.5, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Alok Tom - Dev Gym'. The system tray at the very bottom shows the date and time as 9:08 AM on 10/05/2023, along with weather information (28°C, Partly cloudy).

Live SQL

FeedbackHelpqbdatanzon@ip.edu.ph

Home

SQL Worksheet

ClearFindActionsSaveRun

My Session

Schema

Quick SQL

My Scripts

My Tutorials

Code Library

1 SELECT * FROM EMPLOYEES

EMPLOYEE_ID	FIRST_NAME	MIDDLE_NAME	LAST_NAME	BIRTH_DATE	SALARY	COMMISSION_PCT	PHONE_NUMBER	EMAIL	HIRE_DATE	DEPARTMENT_ID
100	Deila	Manuel	Daria	11-MAR-75	48000	-	9965318232	dcockerham@gmail.com	14-JAN-02	90
101	Randolph	Jan	Nelson	30-AUG-77	35000	-	99274703786	rwkingfield@gmail.com	24-NOV-01	90
102	Naribelle	Evangelina	Diaper	16-JUL-73	35000	-	997042397	mdiape3@gmail.com	06-NOV-03	90
103	Fannie	Alicia	Joszkowir	21-JUN-74	26800	-	943481932	fjoszkowir@gmail.com	20-NOV-08	70
104	Stearne	Dolores	Wickersley	18-NOV-62	25000	-	97585091462	swickersley@gmail.com	04-AUG-03	70
105	Leticia	Bernila	Fawdry	07-SEP-76	25000	-	93080029705	lfawdry@gmail.com	13-AUG-01	70
106	Dorothea	Evangelina	Aggione	25-NOV-73	24500	-	992269989	daggione7@gmail.com	13-JUL-04	70
107	Kliment	Enrique	Barbie	18-JUN-74	24000	-	97230097452	kbarbie@gmail.com	17-DEC-02	70
108	Grady	Piolo	Barnwell	28-AUG-77	29000	-	99337401945	gbarnwell@gmail.com	12-DEC-03	80
109	Chick	Jericho	Gibbe	15-OCT-70	28300	-	96174087069	cgibbea@gmail.com	03-FEB-06	80
110	Abbe	Dominick	Hartnup	08-JUN-69	28000	-	974070497	whartnup@gmail.com	21-OCT-01	80
111	Briny	Jacinto	Allenson	06-JUN-67	27800	-	950411795	ballenson5@gmail.com	08-OCT-04	80

2023 Oracle Live SQL 23.1.5, running Oracle Database 19c EE Express Edition - 19.0.0.0 Database Documentation - Ask Tom - Dev Gyn
Built with Using Oracle APEX - Privacy - Terms of Use

Live SQL

FeedbackHelpqbdatanzon@ip.edu.ph

Home

SQL Worksheet

ClearFindActionsSaveRun

My Session

Schema

Quick SQL

My Scripts

My Tutorials

Code Library

1 SELECT * FROM EMPLOYEES

100	Deila	Manuel	Daria	11-MAR-75	48000	-	9965318232	dcockerham@gmail.com	14-JAN-02	90
101	Randolph	Jan	Nelson	30-AUG-77	35000	-	99274703786	rwkingfield@gmail.com	24-NOV-01	90
102	Naribelle	Evangelina	Diaper	16-JUL-73	35000	-	997042397	mdiape3@gmail.com	06-NOV-03	90
103	Fannie	Alicia	Joszkowir	21-JUN-74	26800	-	943481932	fjoszkowir@gmail.com	20-NOV-08	70
104	Stearne	Dolores	Wickersley	18-NOV-62	25000	-	97585091462	swickersley@gmail.com	04-AUG-03	70
105	Leticia	Bernila	Fawdry	07-SEP-76	25000	-	93080029705	lfawdry@gmail.com	13-AUG-01	70
106	Dorothea	Evangelina	Aggione	25-NOV-73	24500	-	992269989	daggione7@gmail.com	13-JUL-04	70
107	Kliment	Enrique	Barbie	18-JUN-74	24000	-	97230097452	kbarbie@gmail.com	17-DEC-02	70
108	Grady	Piolo	Barnwell	28-AUG-77	29000	-	99337401945	gbarnwell@gmail.com	12-DEC-03	80
109	Chick	Jericho	Gibbe	15-OCT-70	28300	-	96174087069	cgibbea@gmail.com	03-FEB-06	80
110	Abbe	Dominick	Hartnup	08-JUN-69	28000	-	974070497	whartnup@gmail.com	21-OCT-01	80
111	Briny	Jacinto	Allenson	06-JUN-67	27800	-	950411795	ballenson5@gmail.com	08-OCT-04	80

2023 Oracle Live SQL 23.1.5, running Oracle Database 19c EE Express Edition - 19.0.0.0 Database Documentation - Ask Tom - Dev Gyn
Built with Using Oracle APEX - Privacy - Terms of Use

Live SQL

FeedbackHelpqbdatanzon@ip.edu.ph

Home

SQL Worksheet

ClearFindActionsSaveRun

My Session

Schema

Quick SQL

My Scripts

My Tutorials

Code Library

1 SELECT * FROM EMPLOYEES

100	Deila	Manuel	Daria	11-MAR-75	48000	-	9965318232	dcockerham@gmail.com	14-JAN-02	90
101	Randolph	Jan	Nelson	30-AUG-77	35000	-	99274703786	rwkingfield@gmail.com	24-NOV-01	90
102	Naribelle	Evangelina	Diaper	16-JUL-73	35000	-	997042397	mdiape3@gmail.com	06-NOV-03	90
103	Fannie	Alicia	Joszkowir	21-JUN-74	26800	-	943481932	fjoszkowir@gmail.com	20-NOV-08	70
104	Stearne	Dolores	Wickersley	18-NOV-62	25000	-	97585091462	swickersley@gmail.com	04-AUG-03	70
105	Leticia	Bernila	Fawdry	07-SEP-76	25000	-	93080029705	lfawdry@gmail.com	13-AUG-01	70
106	Dorothea	Evangelina	Aggione	25-NOV-73	24500	-	992269989	daggione7@gmail.com	13-JUL-04	70
107	Kliment	Enrique	Barbie	18-JUN-74	24000	-	97230097452	kbarbie@gmail.com	17-DEC-02	70
108	Grady	Piolo	Barnwell	28-AUG-77	29000	-	99337401945	gbarnwell@gmail.com	12-DEC-03	80
109	Chick	Jericho	Gibbe	15-OCT-70	28300	-	96174087069	cgibbea@gmail.com	03-FEB-06	80
110	Abbe	Dominick	Hartnup	08-JUN-69	28000	-	974070497	whartnup@gmail.com	21-OCT-01	80
111	Briny	Jacinto	Allenson	06-JUN-67	27800	-	950411795	ballenson5@gmail.com	08-OCT-04	80

2023 Oracle Live SQL 23.1.5, running Oracle Database 19c EE Express Edition - 19.0.0.0 Database Documentation - Ask Tom - Dev Gyn
Built with Using Oracle APEX - Privacy - Terms of Use

(The employees table has successfully been created and was able to display the table).

Query for Employees Table

```
CREATE TABLE Employees
(employee_id NUMBER (3) NOT NULL
CONSTRAINT employee_id_pk PRIMARY KEY,
first_name VARCHAR (30) NOT NULL,
middle_name VARCHAR (30) NOT NULL,
last_name VARCHAR (30) NOT NULL,
birth_date DATE NOT NULL,
salary NUMBER (8,2) NOT NULL,
commission_pct NUMBER (5,2),
phone_number NUMBER (11) NOT NULL,
    CONSTRAINT phone_number_uk UNIQUE (phone_number),
email VARCHAR (50) NOT NULL,
    CONSTRAINT email_uk UNIQUE (email),
hire_date DATE NOT NULL,
department_id NUMBER (3) NOT NULL
);
```

Table Name 2: Loan Allocations Table (Table Created)

The screenshot displays the Oracle Live SQL web interface. On the left, a sidebar contains navigation links: Home, SQL Worksheet (selected), My Session, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The main area is titled 'SQL Worksheet' and shows a list of 10 SQL statements. The first statement is the 'CREATE TABLE Loan_Allocations' command, which has been executed successfully. Below the SQL list, a message states 'Table created.' The footer of the interface provides technical details: '2023 Oracle - Live SQL 23.1.5, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym' and 'Built with ♥ using Oracle APEX - Privacy - Terms of Use'. The Windows taskbar at the bottom shows the system time as 8:10 AM on 10/05/2023, with a weather widget indicating 28°C and partly cloudy conditions.

```
1. CREATE TABLE Loan_Allocations (
2.   loan_id NUMBER(7) NOT NULL CONSTRAINT loan_id_pk PRIMARY KEY,
3.   loan_category VARCHAR(50) NOT NULL,
4.   loan_startpayment DATE NOT NULL,
5.   loan_endpayment DATE NOT NULL,
6.   loan_amount NUMBER(8, 2) NOT NULL,
7.   interest_rate NUMBER(2, 2) NOT NULL,
8.   down_payment NUMBER(8, 2) NOT NULL,
9. );
10
```

Table created.

2023 Oracle - Live SQL 23.1.5, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym
Built with ♥ using Oracle APEX - Privacy - Terms of Use

8:10 AM
10/05/2023
28°C Partly cloudy

Live SQL

SQL Worksheet

1 SELECT * FROM LOAN_ALLOCATIONS

LOAN_ID	LOAN_CATEGORY	LOAN_STARTPAYMENT	LOAN_ENDPAYMENT	LOAN_AMOUNT	INTEREST_RATE	DOWN_PAYMENT
3834054	Student	22-DEC-20	22-DEC-21	250000	.07	30000
3104937	Student	14-JAN-20	14-JAN-21	230000	.07	26000
3866302	Payday	30-DEC-19	30-DEC-20	220000	.06	25000
3431586	Auto	30-DEC-18	30-DEC-19	210000	.05	24000
2480497	Personal	02-APR-18	02-APR-19	200000	.08	23000
4050243	Auto	11-JUN-17	11-JUN-18	200000	.05	23000
3070934	Personal	31-AUG-17	31-AUG-18	190000	.08	22000
2255547	Personal	09-JUN-16	09-JUN-17	185000	.08	21500
2029575	Payday	22-FEB-16	22-FEB-17	160000	.06	20000
4078770	Personal	01-JUL-15	01-JUL-16	150000	.08	19000

Download CSV

2023 Oracle - Live SQL 23.1.5, running Oracle Database 19c EE Extreme Perf. 19.17.0.20.0 - Database Documentation - Ask Tom - Dev Gym
Built with ♥ Using Oracle APEX - Privacy - Terms of Use

(The loan allocations table has successfully been created and was able to display the table).

Query for Loan Allocations Table:

```
CREATE TABLE Loan_Allocations (
  loan_id NUMBER(7) NOT NULL CONSTRAINT loan_id_pk PRIMARY KEY,
  loan_category VARCHAR(30) NOT NULL,
  loan_startpayment DATE NOT NULL,
  loan_endpayment DATE NOT NULL,
  loan_amount NUMBER(8, 2) NOT NULL,CREATE
  interest_rate NUMBER(2, 2) NOT NULL,
  down_payment NUMBER(8, 2) NOT NULL
);
```

Table Name 3: Borrows Table (Table Created)

Live SQL

FeedbackHelpqbdalanzon@lip.edu.ph

Home

SQL Worksheet

ClearFindActionsSaveRun

SQL Worksheet

My SessionSchemaQuick SQLMy ScriptsMy TutorialsCode Library

```
1. CREATE TABLE Borrowers
2. (borrower_id NUMBER (7) NOT NULL
3. CONSTRAINT borrower_id_pk PRIMARY KEY,
4. first_name VARCHAR (30) NOT NULL,
5. middle_name VARCHAR (30) NOT NULL,
6. last_name VARCHAR (30) NOT NULL,
7. birth_date DATE NOT NULL,
8. address VARCHAR (70) NOT NULL,
9. phone_number NUMBER (11) NOT NULL,
10. CONSTRAINT phone_number_pk2 UNIQUE (phone_number),
11. monthly_salary NUMBER (9) NOT NULL,
12. loan_id NUMBER (7) NOT NULL,
13. CONSTRAINT loan_id_fk FOREIGN KEY (loan_id)
14. REFERENCES Loan_Allocations (loan_id),
15. employee_id NUMBER (3) NOT NULL,
16. CONSTRAINT employee_id_fk FOREIGN KEY (employee_id)
17. REFERENCES Employees (employee_id),
18. payment_id NUMBER (7) NOT NULL
19. );
```

Table created.

2023 Oracle - Live SQL 23.1.5, running Oracle Database 19c EE Extreme Perf. - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym
Built with Using Oracle APEX - Privacy - Terms of Use

Live SQL

FeedbackHelpqbdalanzon@lip.edu.ph

Home

SQL Worksheet

ClearFindActionsSaveRun

SQL Worksheet

My SessionSchemaQuick SQLMy ScriptsMy TutorialsCode Library

```
1 SELECT * FROM BORROWERS
```

BORROWER_ID	FIRST_NAME	MIDDLE_NAME	LAST_NAME	BIRTH_DATE	ADDRESS	PHONE_NUMBER	MONTHLY_SALARY	LOAN_ID	EMPLOYEE_ID	PAYMENT_ID
7200341	Nettie	Patsy	Wannes	30-JUL-70	0431 Summit Pass, Cubao	9439846057	180000	3834054	125	7358203
6791403	Suzie	Tommi	Brundill	13-FEB-76	5250 Morning Hill, Rial	9028491172	97000	3104937	125	6427088
7721155	Natka	Druci	Duffield	15-JUL-86	5766 Coleman Circle, Quezon	9794706307	95000	3866302	125	7267259
7303864	Aldus	Nicholaus	Underhill	27-NOV-84	65675 Graceland Lane, Rial	9272647794	94500	3431586	126	6298799
6187271	Dunstan	Baird	Yakubovics	13-SEP-81	8002 Larry Avenue, Nakati	9460042009	93000	2480497	126	7015762
6243070	Osbert	Roch	Cawston	03-JAN-84	74 Herrick Center, Taguig	9825837467	92000	4050243	126	7866370
7558915	Wendye	Jourdan	Bridie	03-DEC-74	7 Vahlen Center, Pasig	9705126604	91000	3078934	127	7013875
6516977	Joye	Dani	Ledgard	07-FEB-72	3 Parkside Way, Markina	9930813959	90000	2255547	128	7344981
6853965	Caye	Kippie	Scandrick	07-DEC-83	45 Riverside Trail, Rial	9484816220	87000	2029575	129	6325400
7857214	Lock	Genovera	Cranston	06-SEP-85	3464 Rieder Crossing, Pasig	9831851385	85000	4078770	130	7127952

Download CSV

2023 Oracle - Live SQL 23.1.5, running Oracle Database 19c EE Extreme Perf. - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym
Built with Using Oracle APEX - Privacy - Terms of Use

CPU/MEM 100%3:04 AM10/05/2023

(The borrowers table has successfully been created and was able to display the table).

Query for Borrows Table:

```
CREATE TABLE Borrowers
(borrower_id NUMBER (7) NOT NULL
    CONSTRAINT borrower_id_pk PRIMARY KEY,
first_name VARCHAR (30) NOT NULL,
middle_name VARCHAR (30) NOT NULL,
last_name VARCHAR (30) NOT NULL,
birth_date DATE NOT NULL,
address VARCHAR (70) NOT NULL,
phone_number NUMBER (11) NOT NULL,
    CONSTRAINT phone_number_uk2 UNIQUE (phone_number),
monthly_salary NUMBER (9) NOT NULL,
loan_id NUMBER (7) NOT NULL,
    CONSTRAINT loan_id_fk FOREIGN KEY (loan_id)
    REFERENCES Loan_Allocations (loan_id),
employee_id NUMBER (3) NOT NULL,
    CONSTRAINT employee_id_fk FOREIGN KEY (employee_id)
    REFERENCES Employees(employee_id),
payment_id NUMBER (7) NOT NULL
);
```

Table Name 4: Payments Table (Table Created)

The screenshot displays the 'Live SQL' web application interface. On the left, a sidebar contains navigation links: Home, SQL Worksheet (selected), My Session, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The main area is titled 'SQL Worksheet' and contains a SQL script for creating a 'Payments' table. The script includes constraints for primary, foreign, and unique keys. Below the script, a message states 'Table created.' The footer of the application shows the version '2023 Oracle - Live SQL 23.1.5' and the date '18/09/2023'. The bottom of the image shows a Windows taskbar with various icons and a system tray displaying '28°C Partly cloudy' and '3:12 AM 18/09/2023'.

```
1. CREATE TABLE Payments
2. (payment_id NUMBER (7) NOT NULL
3.   CONSTRAINT payment_id_pk PRIMARY KEY,
4.   loan_id NUMBER (7) NOT NULL,
5.   CONSTRAINT loan_id_fk2 FOREIGN KEY (loan_id)
6.     REFERENCES Loan_Allocations (loan_id),
7.   borrower_id NUMBER (7) NOT NULL,
8.   CONSTRAINT borrower_id_fk FOREIGN KEY (borrower_id)
9.     REFERENCES Borrowers (borrower_id),
10.  reference_id NUMBER (7) NOT NULL,
11.  payment_amount NUMBER (9,2) NOT NULL,
12.  payment_method VARCHAR (30) NOT NULL,
13.  payment_status VARCHAR (30) NOT NULL
14. );
```

Table created.

2023 Oracle - Live SQL 23.1.5, running Oracle Database 19c EE Extreme Perf. 18/09/2023 - Database Documentation - Ask Tom - Dev Gym
Built with ❤️ using Oracle APEX - Privacy - Terms of Use

28°C Partly cloudy 3:12 AM 18/09/2023

Live SQL

SQL Worksheet

1 SELECT * FROM PAYMENTS

PAYMENT_ID	LOAN_ID	BORROWER_ID	REFERENCE_ID	PAYMENT_AMOUNT	PAYMENT_METHOD	PAYMENT_STATUS
7358203	3834854	7200341	5468754	262002	Wire Transfer	Pending
6427088	3104937	6751403	5277711	269877	Cash	Pending
7267259	3866302	7721155	5449579	180204	Cash	Pending
6298799	3431586	7303864	5366993	311414	Debit	Pending
7015762	2480497	6187271	5818379	252586	Cash	Pending
7866370	4050243	6243070	5842400	121018	Credit Card	Paid
7013875	3070934	7554915	5536167	101343	Check	Overdue
7344981	2255547	6516977	5280933	235587	Check	Refunded
6325400	2029575	6853965	5139199	157405	Cash	Cancelled
7127952	4078770	7857214	5264361	246369	Credit Card	Failed

Download CSV

2023 Oracle - Live SQL 23.1.5, running Oracle Database 19c EE Extreme Perf. 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gyn
Built with ❤️ using Oracle APEX - Privacy - Terms of Use

CNY/PHP 10.00%

3:05 AM
10/09/2023

(The payments table has successfully been created and was able to display the table).

Query for Payments Table:

CREATE TABLE Payments

(payment_id NUMBER (7) NOT NULL

CONSTRAINT payment_id_pk PRIMARY KEY,

loan_id NUMBER (7) NOT NULL,

CONSTRAINT loan_id_fk2 FOREIGN KEY (loan_id)

REFERENCES Loan_Allocations (loan_id),

borrower_id NUMBER (7) NOT NULL,

CONSTRAINT borrower_id_fk FOREIGN KEY (borrower_id)

REFERENCES Borrowers (borrower_id),

reference_id NUMBER (7) NOT NULL,

payment_amount NUMBER (8,2) NOT NULL,

payment_method VARCHAR (30) NOT NULL,

payment_status VARCHAR (30) NOT NULL

);

Table Name 5: Payments_History Table (Table Created)

The screenshot shows the Live SQL interface with the SQL Worksheet tab active. The SQL code for creating the Payments_History table is displayed in the editor. Below the code, a message states "Table created." The interface includes a sidebar with navigation options like Home, SQL Worksheet, My Session, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The top bar shows the Live SQL logo and user information. The bottom status bar indicates the database version and system information.

```
1. CREATE TABLE Payments_History
2 (reference_id NUMBER (7) NOT NULL,
3  CONSTRAINT reference_id_pk PRIMARY KEY,
4  payment_id NUMBER (7) NOT NULL,
5  CONSTRAINT payment_id_fk FOREIGN KEY (payment_id)
6  REFERENCES Payments(payment_id),
7  loan_id NUMBER (7) NOT NULL,
8  CONSTRAINT loan_id_fk3 FOREIGN KEY (loan_id)
9  REFERENCES Loan_allocations (loan_id),
10 borrower_id NUMBER (7) NOT NULL,
11 CONSTRAINT borrower_id_fk2 FOREIGN KEY (borrower_id)
12 REFERENCES Borrowers (borrower_id),
13 late_fees NUMBER (9,2),
14 balance_before_payment NUMBER (9,2),
15 balance_after_payment NUMBER (9,2)
16 );
```

Table created.

2023 Oracle - Live SQL 23.1.5, running Oracle Database 19c EE Extreme Perf. - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym
Built with ❤️ using Oracle APEX - Privacy - Terms of Use

The screenshot shows the Live SQL interface with the SQL Worksheet tab active. The SQL code for selecting data from the Payments_History table is displayed in the editor. Below the code, the data is displayed in a table format. The interface includes a sidebar with navigation options like Home, SQL Worksheet, My Session, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The top bar shows the Live SQL logo and user information. The bottom status bar indicates the database version and system information.

```
1. SELECT * FROM PAYMENTS_HISTISTORY
```

REFERENCE_ID	PAYMENT_ID	LOAN_ID	BORROWER_ID	LATE_FEES	BALANCE_BEFORE_PAYMENT	BALANCE_AFTER_PAYMENT
5468754	7358283	3834054	7280341	-	30000	180000
5577711	6427888	3104937	6751483	-	26000	170000
5449579	7267259	3860382	7721155	-	25000	170000
5366993	6298799	3431586	7383864	-	24000	150000
5818379	7015762	2480897	6187271	-	23000	150000
5842488	7866378	4058243	6245878	-	23000	200000
5536167	7013875	3078934	7558915	50000	22500	80000
5288993	7344981	2255547	6516977	-	21500	-
5139199	6325488	2829575	6853965	-	20000	-
5264361	7127952	4878778	7857214	-	-	-

Download CSV

2023 Oracle - Live SQL 23.1.5, running Oracle Database 19c EE Extreme Perf. - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym
Built with ❤️ using Oracle APEX - Privacy - Terms of Use

(The payments history table has successfully been created and was able to display the table).

Query for Payments History:

CREATE TABLE Payments_History

(reference_id NUMBER (7) NOT NULL

CONSTRAINT reference_id_pk PRIMARY KEY,

payment_id NUMBER (7) NOT NULL,

CONSTRAINT payment_id_fk FOREIGN KEY (payment_id)

REFERENCES Payments(payment_id),

loan_id NUMBER (7) NOT NULL,

CONSTRAINT loan_id_fk3 FOREIGN KEY (loan_id)

REFERENCES Loan_Allocations (loan_id),

borrower_id NUMBER (7) NOT NULL,

CONSTRAINT borrower_id_fk2 FOREIGN KEY (borrower_id)

REFERENCES Borrowers (borrower_id),

late_fees NUMBER (8,2),

balance_before_payment NUMBER (8,2),

balance_after_payment NUMBER (8,2)

);

TASK 3: Inserting Records (SQL Statements)

Table Name 1: Employees Table (30 rows inserted)

The screenshot displays an SQL Worksheet interface with a sidebar on the left containing navigation links: Home, SQL Worksheet (selected), My Session, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The main workspace shows 30 lines of SQL code, each starting with 'Insert into Employees' followed by a list of employee details in parentheses. The details include employee_id, first_name, middle_name, last_name, birth_date, salary, commission_pct, phone_number, email, hire_date, and department_id. The employees listed are: 100, 'Deila', 'Ranuel', 'Daria', TO_DATE('03/11/1975', 'NN/DD/YYYY'), 4800, 10, 'Ranulph', 'Jan', 'Nelson', TO_DATE('08/10/1977', 'NN/DD/YYYY'), 35, 100, 'Marbelle', 'Evangelina', 'Olaper', TO_DATE('07/16/1977', 'NN/DD/YYYY'), 100, 'Fannie', 'Alicia', 'Josskowi', TO_DATE('06/21/1974', 'NN/DD/YYYY'), 100, 'Stearne', 'Tolares', 'Nickersley', TO_DATE('11/16/1967', 'NN/DD/YYYY'), 100, 'Leticia', 'Bernila', 'Feadry', TO_DATE('09/07/1976', 'NN/DD/YYYY'), 100, 'Dorothea', 'Evangelina', 'Agglione', TO_DATE('11/25/1973', 'NN/DD/YYYY'), 100, 'Vilmont', 'Enrique', 'Bardele', TO_DATE('06/16/1974', 'NN/DD/YYYY'), 100, 'Grady', 'Pisolo', 'Barnwell', TO_DATE('06/28/1977', 'NN/DD/YYYY'), 2, 100, 'Chick', 'Jericho', 'Gibbe', TO_DATE('10/15/1970', 'NN/DD/YYYY'), 28, 110, 'Abbe', 'Dominick', 'Hartup', TO_DATE('06/06/1969', 'NN/DD/YYYY'), 110, 'Brady', 'Jacinto', 'Alinson', TO_DATE('06/02/1967', 'NN/DD/YYYY'), 110, 'Catha', 'Diana', 'Axford', TO_DATE('01/22/1963', 'NN/DD/YYYY'), 275, 110, 'Vasene', 'Jose', 'Jayne', TO_DATE('11/15/1969', 'NN/DD/YYYY'), 27000, 110, 'Willis', 'Dennis', 'Andrew', TO_DATE('11/14/1971', 'NN/DD/YYYY'), 2, 110, 'Mills', 'Nalis', 'Rudman', TO_DATE('08/18/1967', 'NN/DD/YYYY'), 228, 110, 'Cross', 'John', 'Duny', TO_DATE('09/13/1974', 'NN/DD/YYYY'), 22700, 110, 'Janessa', 'Jericho', 'Runtan', TO_DATE('12/25/1975', 'NN/DD/YYYY'), 110, 'Renee', 'Leticia', 'Hick', TO_DATE('06/06/1973', 'NN/DD/YYYY'), 110.

Below the SQL code, the execution results are shown as a list of 30 lines, each stating '1 row(s) inserted.' The status bar at the bottom indicates '2023 Oracle - Live SQL 23.1.5, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Aik Tom - Dev Gym' and 'Built with ♥ Using Oracle APEX - Privacy - Terms of Use'.

These are SQL insert queries for adding new records to a table named "Employees." Each query represents one record to be added, and the values enclosed in parentheses represent the values for each field in the record.

Query for Inserting Employees Data:

```
insert into Employees (employee_id, first_name, middle_name, last_name, birth_date, salary,
commission_pct, phone_number, email, hire_date, department_id) values (100, 'Delia', 'Manuel', 'Daria',
TO_DATE('03/11/1975', 'MM/DD/YYYY'), 40000.00, NULL, '09965318232', 'dcockerham1@gmail.com',
TO_DATE('01/14/2002', 'MM/DD/YYYY'), 90);
insert into Employees (employee_id, first_name, middle_name, last_name, birth_date, salary,
commission_pct, phone_number, email, hire_date, department_id) values (101, 'Randolph', 'Jan',
'Melzon', TO_DATE('08/30/1977', 'MM/DD/YYYY'), 35000.00, NULL, '099271703786',
'rwingfield2@gmail.com', TO_DATE('11/24/2001', 'MM/DD/YYYY'), 90);
insert into Employees (employee_id, first_name, middle_name, last_name, birth_date, salary,
commission_pct, phone_number, email, hire_date, department_id) values (102, 'Maribelle', 'Evangolina',
'Diaper', TO_DATE('07/16/1973', 'MM/DD/YYYY'), 35000.00, NULL, '0997042397',
'mdiaper3@gmail.com', TO_DATE('11/06/2003', 'MM/DD/YYYY'), 90);
insert into Employees (employee_id, first_name, middle_name, last_name, birth_date, salary,
commission_pct, phone_number, email, hire_date, department_id) values (103, 'Fannie', 'Alicia',
'Josskoviz', TO_DATE('06/21/1974', 'MM/DD/YYYY'), 26800.00, NULL, '0943481932',
'fjosskoviz4@gmail.com', TO_DATE('11/20/1998', 'MM/DD/YYYY'), 70);
insert into Employees (employee_id, first_name, middle_name, last_name, birth_date, salary,
commission_pct, phone_number, email, hire_date, department_id) VALUES (104, 'Stearne', 'Dolores',
'Wickersley', TO_DATE('11/18/1962', 'MM/DD/YYYY'), 25000.00, NULL, '097585091462',
'swickersley0@gmail.com', TO_DATE('08/04/2003', 'MM/DD/YYYY'), 70);
insert into Employees (employee_id, first_name, middle_name, last_name, birth_date, salary,
commission_pct, phone_number, email, hire_date, department_id) VALUES (105, 'Leticia', 'Bernila',
'Fawdry', TO_DATE('09/07/1976', 'MM/DD/YYYY'), 25000.00, NULL, '093080029705',
'lfawdry6@gmail.com', TO_DATE('08/13/2001', 'MM/DD/YYYY'), 70);
insert into Employees (employee_id, first_name, middle_name, last_name, birth_date, salary,
commission_pct, phone_number, email, hire_date, department_id) VALUES (106, 'Dorothea',
'Evangolina', 'Agglione', TO_DATE('11/25/1973', 'MM/DD/YYYY'), 24500.00, NULL, '0992269989',
'dagglione7@gmail.com', TO_DATE('07/13/2004', 'MM/DD/YYYY'), 70);
insert into Employees (employee_id, first_name, middle_name, last_name, birth_date, salary,
commission_pct, phone_number, email, hire_date, department_id) VALUES (107, 'Kliment', 'Enrique',
'Barbie', TO_DATE('06/18/1974', 'MM/DD/YYYY'), 24000.00, NULL, '097230097452',
'kbarbie8@gmail.com', TO_DATE('12/17/2002', 'MM/DD/YYYY'), 70);
insert into Employees (employee_id, first_name, middle_name, last_name, birth_date, salary,
commission_pct, phone_number, email, hire_date, department_id) values (108, 'Grady', 'Piolo',
'Barnwell', TO_DATE('08/28/1977', 'MM/DD/YYYY'), 29000.00, NULL, '099337401945',
'gbarnwell9@gmail.com', TO_DATE('12/12/2003', 'MM/DD/YYYY'), 80);
insert into Employees (employee_id, first_name, middle_name, last_name, birth_date, salary,
commission_pct, phone_number, email, hire_date, department_id) values (109, 'Chick', 'Jericho',
'Gibbe', TO_DATE('10/15/1970', 'MM/DD/YYYY'), 28300.00, NULL, '096174087069',
'cgibbea@gmail.com', TO_DATE('02/03/1998', 'MM/DD/YYYY'), 80);
```



```

insert into Employees (employee_id, first_name, middle_name, last_name, birth_date, salary,
commission_pct, phone_number, email, hire_date, department_id) values (110, 'Abbe', 'Dominick',
'Hartnup', TO_DATE('06/08/1969', 'MM/DD/YYYY'), 28000.00, NULL, '0974070487',
'ahartnupb@gmail.com', TO_DATE('10/21/2001', 'MM/DD/YYYY'),80);
insert into Employees (employee_id, first_name, middle_name, last_name, birth_date, salary,
commission_pct, phone_number, email, hire_date, department_id) values (111, 'Briny', 'Jacinto',
'Allenson', TO_DATE('06/06/1967', 'MM/DD/YYYY'), 27800.00, NULL, '0950441795',
'ballenson5@gmail.com', TO_DATE('10/08/2004', 'MM/DD/YYYY'),80);
insert into Employees (employee_id, first_name, middle_name, last_name, birth_date, salary,
commission_pct, phone_number, email, hire_date, department_id) values (112, 'Catha', 'Diana', 'Axford',
TO_DATE('01/22/1963', 'MM/DD/YYYY'), 27500.00, NULL, '090653593237', 'caxfordr@gmail.com',
TO_DATE('08/10/2003', 'MM/DD/YYYY'),80);
insert into Employees (employee_id, first_name, middle_name, last_name, birth_date, salary,
commission_pct, phone_number, email, hire_date, department_id) values (113, 'Keene', 'Jose', 'Jayme',
TO_DATE('11/15/1969', 'MM/DD/YYYY'), 27000.00, NULL, '0926762522', 'kjaymes@gmail.com',
TO_DATE('06/14/2001', 'MM/DD/YYYY'),80);
insert into Employees (employee_id, first_name, middle_name, last_name, birth_date, salary,
commission_pct, phone_number, email, hire_date, department_id) VALUES (114, 'Willi', 'Dennis',
'Andreas', TO_DATE('11/14/1972', 'MM/DD/YYYY'), 23000.00, NULL, '094194420118',
'wandreasc@gmail.com', TO_DATE('04/14/2002', 'MM/DD/YYYY'),40);
insert into Employees (employee_id, first_name, middle_name, last_name, birth_date, salary,
commission_pct, phone_number, email, hire_date, department_id) VALUES (115, 'Hilly', 'Malis',
'Rudham', TO_DATE('08/10/1967', 'MM/DD/YYYY'), 22800.00, NULL, '09760036983',
'hrudhamd@gmail.com', TO_DATE('07/15/2002', 'MM/DD/YYYY'),40);
insert into Employees (employee_id, first_name, middle_name, last_name, birth_date, salary,
commission_pct, phone_number, email, hire_date, department_id) VALUES (116, 'Cross', 'John', 'Duny',
TO_DATE('05/13/1974', 'MM/DD/YYYY'), 22700.00, NULL, '09539357059', 'cdunye@gmail.com',
TO_DATE('12/09/2005', 'MM/DD/YYYY'),40);
insert into Employees (employee_id, first_name, middle_name, last_name, birth_date, salary,
commission_pct, phone_number, email, hire_date, department_id) VALUES (117, 'Janessa', 'Jericho',
'Bunton', TO_DATE('12/25/1975', 'MM/DD/YYYY'), 22700.00, NULL, '09338823761',
'jbuntonf@gmail.com', TO_DATE('06/28/2002', 'MM/DD/YYYY'), 40);
insert into Employees (employee_id, first_name, middle_name, last_name, birth_date, salary,
commission_pct, phone_number, email, hire_date, department_id) VALUES (119, 'Ella', 'Faurillo', 'Likly',
TO_DATE('09/22/1963', 'MM/DD/YYYY'), 22600.00, NULL, '095042294296', 'eliklyf@gmail.com',
TO_DATE('03/26/2004', 'MM/DD/YYYY'), 40);
insert into Employees (employee_id, first_name, middle_name, last_name, birth_date, salary,
commission_pct, phone_number, email, hire_date, department_id) values (120, 'Jacynth', 'Lou',
'Benthall', TO_DATE('06/16/1964', 'MM/DD/YYYY'), 22500.00, NULL, '09778073708',
'jbenthalli@gmail.com', TO_DATE('10/03/2004', 'MM/DD/YYYY'), 40);
insert into Employees (employee_id, first_name, middle_name, last_name, birth_date, salary,
commission_pct, phone_number, email, hire_date, department_id) values (121, 'Towny', 'Mausy',
'Roostan', TO_DATE('01/03/1963', 'MM/DD/YYYY'), 24000.00, NULL, '095756607353',
'troostanj@gmail.com', TO_DATE('05/26/2000', 'MM/DD/YYYY'), 50);
insert into Employees (employee_id, first_name, middle_name, last_name, birth_date, salary,
commission_pct, phone_number, email, hire_date, department_id) values (122, 'Gwendolin', 'Piolo',

```

```

'Montan', TO_DATE('04/25/1962', 'MM/DD/YYYY'), 23600.00, NULL, '096494092534',
'gwarboysk@gmail.com', TO_DATE('11/01/2001', 'MM/DD/YYYY'), 50);
insert into Employees (employee_id, first_name, middle_name, last_name, birth_date, salary,
commission_pct, phone_number, email, hire_date, department_id) VALUES (123, 'Borkum', 'Paulo',
'Garron', TO_DATE('08/02/1961', 'MM/DD/YYYY'), 23500.00, NULL, '0956921844',
'ggarronl@gmail.com', TO_DATE('03/17/2003', 'MM/DD/YYYY'), 50);
insert into Employees (employee_id, first_name, middle_name, last_name, birth_date, salary,
commission_pct, phone_number, email, hire_date, department_id) VALUES (124, 'Drew', 'Jose',
'Gimbrett', TO_DATE('05/31/1963', 'MM/DD/YYYY'), 23400.00, NULL, '097963407598',
'dgimbrettm@gmail.com', TO_DATE('08/03/2001', 'MM/DD/YYYY'), 50);
insert into Employees (employee_id, first_name, middle_name, last_name, birth_date, salary,
commission_pct, phone_number, email, hire_date, department_id) VALUES (125, 'Kerianne', 'Jericho',
'Nicholas', TO_DATE('06/04/1969', 'MM/DD/YYYY'), 23400.00, NULL, '0983772851',
'knicholasn@gmail.com', TO_DATE('04/21/2002', 'MM/DD/YYYY'), 50);
insert into Employees (employee_id, first_name, middle_name, last_name, birth_date, salary,
commission_pct, phone_number, email, hire_date, department_id) VALUES (126, 'Evey', 'Jacinto',
'Gonzaga', TO_DATE('12/02/1970', 'MM/DD/YYYY'), 87000.00, 0.4, '0982690984',
'ebraineo@gmail.com', TO_DATE('01/25/1992', 'MM/DD/YYYY'), 100);
insert into Employees (employee_id, first_name, middle_name, last_name, birth_date, salary,
commission_pct, phone_number, email, hire_date, department_id) VALUES (127, 'Elladine', 'Jericho',
'Lima', TO_DATE('02/29/1972', 'MM/DD/YYYY'), 85000.00, 0.4, '0957500209',
'elosseljongp@gmail.com', TO_DATE('12/03/1994', 'MM/DD/YYYY'), 100);
insert into Employees (employee_id, first_name, middle_name, last_name, birth_date, salary,
commission_pct, phone_number, email, hire_date, department_id) VALUES (128, 'Laney', 'Divina',
'Obatay', TO_DATE('01/11/1972', 'MM/DD/YYYY'), 80000.00, 0.3, '0968153024',
'lhamfleetq@gmail.com', TO_DATE('06/12/1993', 'MM/DD/YYYY'), 100);
insert into Employees (employee_id, first_name, middle_name, last_name, birth_date, salary,
commission_pct, phone_number, email, hire_date, department_id) values (129, 'Dulcine', 'Ferdinand',
'Talby', TO_DATE('11/23/1961', 'MM/DD/YYYY'), 79000.00, 0.3, '09689209912', 'dtalbyt@gmail.com',
TO_DATE('12/15/1996', 'MM/DD/YYYY'), 100);
insert into Employees (employee_id, first_name, middle_name, last_name, birth_date, salary,
commission_pct, phone_number, email, hire_date, department_id) values (130, 'Dalzayn', 'Ferrucio',
'Gabbie', TO_DATE('11/21/1966', 'MM/DD/YYYY'), 75000.00, 0.2, '0964153024', 'dltgb@gmail.com',
TO_DATE('02/25/1998', 'MM/DD/YYYY'), 100);

```

Table Name 2: Loan_Allocations Table (10 rows inserted)

```
1 Insert Into Loan_Allocations (loan_id, loan_category, loan_startpayment, loan_endpayment, loan_amount, interest_rate, down_payment) VALUES (3834054, 'Student', TO_DATE('12/22/2020', 'MM/DD/YYYY'), TO_DATE('12/22/2021', 'MM/DD/YYYY'), 250000.00, 0.07, 30000.00);
2 Insert Into Loan_Allocations (loan_id, loan_category, loan_startpayment, loan_endpayment, loan_amount, interest_rate, down_payment) VALUES (3104937, 'Student', TO_DATE('02/14/2020', 'MM/DD/YYYY'), TO_DATE('02/14/2021', 'MM/DD/YYYY'), 210000.00, 0.07, 21000.00);
3 Insert Into Loan_Allocations (loan_id, loan_category, loan_startpayment, loan_endpayment, loan_amount, interest_rate, down_payment) VALUES (3860302, 'Payday', TO_DATE('12/30/2019', 'MM/DD/YYYY'), TO_DATE('12/30/2020', 'MM/DD/YYYY'), 210000.00, 0.07, 21000.00);
4 Insert Into Loan_Allocations (loan_id, loan_category, loan_startpayment, loan_endpayment, loan_amount, interest_rate, down_payment) VALUES (3431586, 'Auto', TO_DATE('12/30/2018', 'MM/DD/YYYY'), TO_DATE('12/30/2019', 'MM/DD/YYYY'), 210000.00, 0.07, 21000.00);
5 Insert Into Loan_Allocations (loan_id, loan_category, loan_startpayment, loan_endpayment, loan_amount, interest_rate, down_payment) VALUES (2480597, 'Personal', TO_DATE('06/02/2018', 'MM/DD/YYYY'), TO_DATE('06/02/2019', 'MM/DD/YYYY'), 200000.00, 0.07, 20000.00);
6 Insert Into Loan_Allocations (loan_id, loan_category, loan_startpayment, loan_endpayment, loan_amount, interest_rate, down_payment) VALUES (4050243, 'Auto', TO_DATE('06/11/2017', 'MM/DD/YYYY'), TO_DATE('06/11/2018', 'MM/DD/YYYY'), 200000.00, 0.07, 20000.00);
7 Insert Into Loan_Allocations (loan_id, loan_category, loan_startpayment, loan_endpayment, loan_amount, interest_rate, down_payment) VALUES (3070934, 'Personal', TO_DATE('06/31/2013', 'MM/DD/YYYY'), TO_DATE('06/31/2014', 'MM/DD/YYYY'), 150000.00, 0.07, 15000.00);
8 Insert Into Loan_Allocations (loan_id, loan_category, loan_startpayment, loan_endpayment, loan_amount, interest_rate, down_payment) VALUES (2255547, 'Personal', TO_DATE('06/09/2016', 'MM/DD/YYYY'), TO_DATE('06/09/2017', 'MM/DD/YYYY'), 185000.00, 0.07, 18500.00);
9 Insert Into Loan_Allocations (loan_id, loan_category, loan_startpayment, loan_endpayment, loan_amount, interest_rate, down_payment) VALUES (2029579, 'Payday', TO_DATE('02/22/2016', 'MM/DD/YYYY'), TO_DATE('02/22/2017', 'MM/DD/YYYY'), 160000.00, 0.07, 16000.00);
10 Insert Into Loan_Allocations (loan_id, loan_category, loan_startpayment, loan_endpayment, loan_amount, interest_rate, down_payment) VALUES (4070770, 'Personal', TO_DATE('07/01/2015', 'MM/DD/YYYY'), TO_DATE('07/01/2016', 'MM/DD/YYYY'), 150000.00, 0.07, 15000.00);
```

These SQL insert queries are used to add new records into the "Loan_Allocations" table with the following columns: loan_id, loan_category, loan_startpayment, loan_endpayment, loan_amount, interest_rate, and down_payment. Each insert statement is inserting a new row into the table with the corresponding values provided.

- The "Loan_Allocations" table is used to store information about loans that have been allocated to borrowers. This table typically includes columns such as loan ID, loan category, loan start and end dates, loan amount, interest rate, and down payment.
- To add new loan allocation records to the table, SQL insert queries are used. These queries specify the columns being inserted into (loan_id, loan_category, loan_startpayment, loan_endpayment, loan_amount, interest_rate, and down_payment) and the values to be inserted into each column for a new row.
- Each insert statement inserts a new row into the "Loan_Allocations" table with the corresponding values provided. This allows loan administrators to track loan allocations and manage payments over time.

Query for Inserting Loan Allocations data:

```
insert into Loan_Allocations (loan_id, loan_category, loan_startpayment, loan_endpayment, loan_amount, interest_rate, down_payment) VALUES (3834054, 'Student', TO_DATE('12/22/2020', 'MM/DD/YYYY'), TO_DATE('12/22/2021', 'MM/DD/YYYY'), 250000.00, 0.07, 30000.00);
```

```
insert into Loan_Allocations (loan_id, loan_category, loan_startpayment, loan_endpayment,
loan_amount, interest_rate, down_payment) values (3104937, 'Student', TO_DATE('01/14/2020',
'MM/DD/YYYY'), TO_DATE('01/14/2021', 'MM/DD/YYYY'), 230000.00, 0.07, 26000.00);
insert into Loan_Allocations (loan_id, loan_category, loan_startpayment, loan_endpayment,
loan_amount, interest_rate, down_payment) values (3866302, 'Payday', TO_DATE('12/30/2019',
'MM/DD/YYYY'), TO_DATE('12/30/2020', 'MM/DD/YYYY'), 220000.00, 0.06, 25000.00);
insert into Loan_Allocations (loan_id, loan_category, loan_startpayment, loan_endpayment,
loan_amount, interest_rate, down_payment) values (3431586, 'Auto', TO_DATE('12/30/2018',
'MM/DD/YYYY'), TO_DATE('12/30/2019', 'MM/DD/YYYY'), 210000.00, 0.05, 24000.00);
insert into Loan_Allocations (loan_id, loan_category, loan_startpayment, loan_endpayment,
loan_amount, interest_rate, down_payment) values (2480497, 'Personal', TO_DATE('04/02/2018',
'MM/DD/YYYY'), TO_DATE('04/02/2019', 'MM/DD/YYYY'), 200000.00, 0.08, 23000.00);
insert into Loan_Allocations (loan_id, loan_category, loan_startpayment, loan_endpayment,
loan_amount, interest_rate, down_payment) values (4050243, 'Auto', TO_DATE('06/11/2017',
'MM/DD/YYYY'), TO_DATE('06/11/2018', 'MM/DD/YYYY'), 200000.00, 0.05, 23000.00);
insert into Loan_Allocations (loan_id, loan_category, loan_startpayment, loan_endpayment,
loan_amount, interest_rate, down_payment) values (3070934, 'Personal', TO_DATE('08/31/2017',
'MM/DD/YYYY'), TO_DATE('08/31/2018', 'MM/DD/YYYY'), 190000.00, 0.08, 22000.00);
insert into Loan_Allocations (loan_id, loan_category, loan_startpayment, loan_endpayment,
loan_amount, interest_rate, down_payment) values (2255547, 'Personal', TO_DATE('06/09/2016',
'MM/DD/YYYY'), TO_DATE('06/09/2017', 'MM/DD/YYYY'), 185000.00, 0.08, 21500.00);
insert into Loan_Allocations (loan_id, loan_category, loan_startpayment, loan_endpayment,
loan_amount, interest_rate, down_payment) values (2029575, 'Payday', TO_DATE('02/22/2016',
'MM/DD/YYYY'), TO_DATE('02/22/2017', 'MM/DD/YYYY'), 160000.00, 0.06, 20000.00);
insert into Loan_Allocations (loan_id, loan_category, loan_startpayment, loan_endpayment,
loan_amount, interest_rate, down_payment) values (4078770, 'Personal', TO_DATE('07/01/2015',
'MM/DD/YYYY'), TO_DATE('07/01/2016', 'MM/DD/YYYY'), 150000.00, 0.08, 19000.00);
```

Table Name 3: Borrowers Table (10 rows inserted)

```
1 --(Out of 30 employees, we use 5 employees that has department_id of 100 which are the ones that approved the loan application.)
2 Insert Into Borrowers (borrower_id, first_name, middle_name, last_name, birth_date, address, phone_number, monthly_salary, loan_id, employee_id, payment_id) values (7208041, 'Bettie', 'Patsy', 'Wanner', TO_DATE('07/30/1970', 'MM/DD/YYYY'), 'NN/CO/YYYY')
3 Insert Into Borrowers (borrower_id, first_name, middle_name, last_name, birth_date, address, phone_number, monthly_salary, loan_id, employee_id, payment_id) values (6751480, 'Suzie', 'Rami', 'Brundill', TO_DATE('02/13/1976', 'MM/DD/YYYY'), 'NN/CO/YYYY')
4 Insert Into Borrowers (borrower_id, first_name, middle_name, last_name, birth_date, address, phone_number, monthly_salary, loan_id, employee_id, payment_id) values (7721155, 'Nakea', 'Traci', 'Duffield', TO_DATE('07/15/1986', 'MM/DD/YYYY'), 'NN/CO/YYYY')
5 Insert Into Borrowers (borrower_id, first_name, middle_name, last_name, birth_date, address, phone_number, monthly_salary, loan_id, employee_id, payment_id) values (7308464, 'Adrian', 'Nicolaius', 'Underhill', TO_DATE('11/27/1984', 'MM/DD/YYYY'), 'NN/CO/YYYY')
6 Insert Into Borrowers (borrower_id, first_name, middle_name, last_name, birth_date, address, phone_number, monthly_salary, loan_id, employee_id, payment_id) values (6187271, 'Dunstan', 'Isid', 'Yakovovic', TO_DATE('09/13/1981', 'MM/DD/YYYY'), 'NN/CO/YYYY')
7 Insert Into Borrowers (borrower_id, first_name, middle_name, last_name, birth_date, address, phone_number, monthly_salary, loan_id, employee_id, payment_id) values (6343070, 'Osbert', 'Roch', 'Cawston', TO_DATE('01/03/1984', 'MM/DD/YYYY'), 'NN/CO/YYYY')
8 Insert Into Borrowers (borrower_id, first_name, middle_name, last_name, birth_date, address, phone_number, monthly_salary, loan_id, employee_id, payment_id) values (7558715, 'Wendye', 'Jaurdan', 'Bridle', TO_DATE('11/03/1974', 'MM/DD/YYYY'), 'NN/CO/YYYY')
9 Insert Into Borrowers (borrower_id, first_name, middle_name, last_name, birth_date, address, phone_number, monthly_salary, loan_id, employee_id, payment_id) values (6516077, 'Jope', 'Saul', 'Lefgard', TO_DATE('02/07/1972', 'MM/DD/YYYY'), 'NN/CO/YYYY')
10 Insert Into Borrowers (borrower_id, first_name, middle_name, last_name, birth_date, address, phone_number, monthly_salary, loan_id, employee_id, payment_id) values (6853965, 'Caye', 'Kippie', 'Scandrick', TO_DATE('12/07/1983', 'MM/DD/YYYY'), 'NN/CO/YYYY')
11 Insert Into Borrowers (borrower_id, first_name, middle_name, last_name, birth_date, address, phone_number, monthly_salary, loan_id, employee_id, payment_id) values (7857214, 'Lock', 'Genevera', 'Cranston', TO_DATE('09/26/1965', 'MM/DD/YYYY'), 'NN/CO/YYYY')
12
```

1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.

2023 Oracle - Use SQL 23.1.5, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gm
Built with ♥ Using Oracle APEX - Privacy - Terms of Use

These are SQL INSERT queries that insert data into a table named "Borrowers". The table has 10 columns, namely borrower_id, first_name, middle_name, last_name, birth_date, address, phone_number, monthly_salary, loan_id, employee_id, and payment_id.

- The borrower table stores information about each borrower who applies for a loan, including their unique borrower ID, name, birth date, address, phone number, and monthly salary. This table is used to track borrower information and ensure that loans are only granted to eligible borrowers.
- The loan table stores information about each loan that is granted to a borrower, including the loan ID, borrower ID, employee ID (who approved the loan), and any associated payments. This table is used to track loan details and ensure that loans are properly approved and managed.
- The payment table stores information about each payment made towards a loan, including the payment ID, loan ID, and amount paid. This table is used to track payments and ensure that borrowers are making payments on time.
- The employee table stores information about each employee who works in the loan management system, including their unique employee ID, name, and any associated loans they have approved. This table is used to track employee information and ensure that loans are properly approved and managed.

Query for Inserting datafor Borrows Table:

insert into Borrowers (borrower_id, first_name, middle_name, last_name, birth_date, address, phone_number, monthly_salary, loan_id, employee_id, payment_id) values (7200341, 'Nettie', 'Patsy', 'Wannes', TO_DATE('07/30/1970', 'MM/DD/YYYY'), '0431 Summit Pass, Cubao', '09439846057', 108000.00, 3834054, 125, 7358203);

insert into Borrowers (borrower_id, first_name, middle_name, last_name, birth_date, address, phone_number, monthly_salary, loan_id, employee_id, payment_id) values (6751403, 'Suzie', 'Tommi', 'Brundill', TO_DATE('02/13/1976', 'MM/DD/YYYY'), '5250 Morning Hill, Rizal', '09028891172', 97000.00, 3104937, 125, 6427088);

insert into Borrowers (borrower_id, first_name, middle_name, last_name, birth_date, address, phone_number, monthly_salary, loan_id, employee_id, payment_id) values (7721155, 'Natka', 'Druci', 'Duffield', TO_DATE('07/15/1986', 'MM/DD/YYYY'), '5766 Coleman Circle, Quezon', '09794706307', 95000.00, 3866302, 125, 7267259);

insert into Borrowers (borrower_id, first_name, middle_name, last_name, birth_date, address, phone_number, monthly_salary, loan_id, employee_id, payment_id) values (7303864, 'Aldus', 'Nickolaus', 'Underhill', TO_DATE('11/27/1984', 'MM/DD/YYYY'), '65675 Graceland Lane, Rizal', '09272647794', 94500.00, 3431586, 126, 6298799);

insert into Borrowers (borrower_id, first_name, middle_name, last_name, birth_date, address, phone_number, monthly_salary, loan_id, employee_id, payment_id) values (6187271, 'Dunstan', 'Baird', 'Yakubovics', TO_DATE('09/13/1981', 'MM/DD/YYYY'), '8002 Larry Avenue, Makati', '09460042009', 93000.00, 2480497, 126, 7015762);

insert into Borrowers (borrower_id, first_name, middle_name, last_name, birth_date, address, phone_number, monthly_salary, loan_id, employee_id, payment_id) values (6243070, 'Osbert', 'Roch', 'Cawston', TO_DATE('01/03/1984', 'MM/DD/YYYY'), '74 Merrick Center, Taguig', '09825837467', 92000.00, 4050243, 126, 7866370);

insert into Borrowers (borrower_id, first_name, middle_name, last_name, birth_date, address, phone_number, monthly_salary, loan_id, employee_id, payment_id) values (7558915, 'Wendye', 'Jourdan', 'Bridie', TO_DATE('12/03/1974', 'MM/DD/YYYY'), '7 Vahlen Center, Pasig', '09765126604', 91000.00, 3070934, 127, 7013875);

insert into Borrowers (borrower_id, first_name, middle_name, last_name, birth_date, address, phone_number, monthly_salary, loan_id, employee_id, payment_id) values (6516977, 'Joye', 'Dani', 'Ledgard', TO_DATE('02/07/1972', 'MM/DD/YYYY'), '3 Parkside Way, marikina', '09930813959', 90000.00, 2255547, 128, 7344981);

insert into Borrowers (borrower_id, first_name, middle_name, last_name, birth_date, address, phone_number, monthly_salary, loan_id, employee_id, payment_id) values (6853965, 'Caye', 'Kippie', 'Scandrick', TO_DATE('12/07/1983', 'MM/DD/YYYY'), '45 Riverside Trail, Rizal', '09484816220', 87000.00, 2029575, 129, 6325400);

insert into Borrowers (borrower_id, first_name, middle_name, last_name, birth_date, address, phone_number, monthly_salary, loan_id, employee_id, payment_id) values (7857214, 'Lock', 'Genovera', 'Cranston', TO_DATE('09/06/1985', 'MM/DD/YYYY'), '3464 Rieder Crossing, Pasig', '09831851385', 85000.00, 4078770, 130, 7127952);

Table Name 4: Payments Table (10 rows inserted)

The screenshot shows a web-based SQL editor interface. On the left is a sidebar with navigation links: Home, SQL Worksheet, My Session, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The main area is titled 'SQL Worksheet' and contains 10 numbered SQL insert statements. Each statement follows the pattern: `Insert Into Payments (payment_id, loan_id, borrower_id, reference_id, payment_amount, payment_method, payment_status) values (...)`. The values include various numeric IDs and strings for payment methods and statuses. Below the queries, the execution results are shown as a list of 10 messages, each stating '1 row(s) inserted.' At the bottom of the interface, a status bar indicates the database version: '2023 Oracle - Live SQL 23.1.5, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym'.

```
1 Insert Into Payments (payment_id, loan_id, borrower_id, reference_id, payment_amount, payment_method, payment_status) values (7358289, 5834824, 7280941, 5468754, 262002, 'Wire Transfer', 'Pending');
2 Insert Into Payments (payment_id, loan_id, borrower_id, reference_id, payment_amount, payment_method, payment_status) values (6657088, 5180693, 6751448, 5577713, 269877, 'Cash', 'Pending');
3 Insert Into Payments (payment_id, loan_id, borrower_id, reference_id, payment_amount, payment_method, payment_status) values (7267259, 3866382, 7721155, 5449579, 180208, 'Cash', 'Pending');
4 Insert Into Payments (payment_id, loan_id, borrower_id, reference_id, payment_amount, payment_method, payment_status) values (6298799, 5431586, 7303864, 5366993, 311414, 'Debit', 'Pending');
5 Insert Into Payments (payment_id, loan_id, borrower_id, reference_id, payment_amount, payment_method, payment_status) values (7855763, 5484897, 6187271, 5818379, 255266, 'Cash', 'Pending');
6 Insert Into Payments (payment_id, loan_id, borrower_id, reference_id, payment_amount, payment_method, payment_status) values (7866378, 4058243, 6243878, 5842488, 121818, 'Credit Card', 'Paid');
7 Insert Into Payments (payment_id, loan_id, borrower_id, reference_id, payment_amount, payment_method, payment_status) values (7813875, 3878934, 7558915, 5536167, 181343, 'Check', 'Overdue');
8 Insert Into Payments (payment_id, loan_id, borrower_id, reference_id, payment_amount, payment_method, payment_status) values (7344081, 2255547, 6516977, 5288993, 235587, 'Check', 'Refunded');
9 Insert Into Payments (payment_id, loan_id, borrower_id, reference_id, payment_amount, payment_method, payment_status) values (6325488, 2029575, 6853905, 5139199, 157485, 'Cash', 'Cancelled');
10 Insert Into Payments (payment_id, loan_id, borrower_id, reference_id, payment_amount, payment_method, payment_status) values (7127952, 4878778, 7857214, 5164361, 248368, 'Credit Card', 'Failed');
```

1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.

2023 Oracle - Live SQL 23.1.5, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym
Built with ♥ Using Oracle APEX - Privacy - Terms of Use

These are SQL queries to insert data into the Payments table. Each query inserts a single row into the table, with the specified values for the columns. Here's a breakdown of what each column represents:

- **payment_id**: a unique identifier for the payment being made.
- **loan_id**: the ID of the loan associated with this payment.
- **borrower_id**: the ID of the borrower making the payment.
- **reference_id**: an ID or reference number associated with the payment (e.g. an invoice number).
- **payment_amount**: the amount of the payment, in the currency of the loan.
- **payment_method**: the method used to make the payment (e.g. wire transfer, cash, debit, credit card, check).
- **payment_status**: the current status of the payment (e.g. pending, paid, overdue, refunded, canceled, failed).

So, each of these queries is inserting a payment into the Payments table with a unique **payment_id** and the specified values for the other columns.

Query for Inserting Data in Payments Table:

```
insert into Payments (payment_id, loan_id, borrower_id, reference_id, payment_amount,
payment_method, payment_status) values (7358203, 3834054, 7200341, 5468754, 262002, 'Wire
Transfer', 'Pending');
insert into Payments (payment_id, loan_id, borrower_id, reference_id, payment_amount,
payment_method, payment_status) values (6427088, 3104937, 6751403, 5577711, 269877, 'Cash',
'Pending');
insert into Payments (payment_id, loan_id, borrower_id, reference_id, payment_amount,
payment_method, payment_status) values (7267259, 3866302, 7721155, 5449579, 180204, 'Cash',
'Pending');
insert into Payments (payment_id, loan_id, borrower_id, reference_id, payment_amount,
payment_method, payment_status) values (6298799, 3431586, 7303864, 5366993, 311414, 'Debit',
'Pending');
insert into Payments (payment_id, loan_id, borrower_id, reference_id, payment_amount,
payment_method, payment_status) values (7015762, 2480497, 6187271, 5818379, 252586, 'Cash',
'Pending');
insert into Payments (payment_id, loan_id, borrower_id, reference_id, payment_amount,
payment_method, payment_status) values (7866370, 4050243, 6243070, 5842400, 121018, 'Credit
Card', 'Paid');
insert into Payments (payment_id, loan_id, borrower_id, reference_id, payment_amount,
payment_method, payment_status) values (7013875, 3070934, 7558915, 5536167, 101343, 'Check',
'Overdue');
insert into Payments (payment_id, loan_id, borrower_id, reference_id, payment_amount,
payment_method, payment_status) values (7344981, 2255547, 6516977, 5280933, 235587, 'Check',
'Refunded');
insert into Payments (payment_id, loan_id, borrower_id, reference_id, payment_amount,
payment_method, payment_status) values (6325400, 2029575, 6853965, 5139199, 157405, 'Cash',
'Cancelled');
insert into Payments (payment_id, loan_id, borrower_id, reference_id, payment_amount,
payment_method, payment_status) values (7127952, 4078770, 7857214, 5264361, 248369, 'Credit
Card', 'Failed');
```


Table Name 5: Payments_History Table (10 rows inserted)

The screenshot shows a web-based SQL editor with a sidebar on the left containing links to Home, SQL Worksheet, My Session, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The main area displays 10 INSERT statements for the Payments_History table, each with a unique set of values for the columns: reference_id, payment_id, loan_id, borrower_id, late_fees, balance_before_payment, and balance_after_payment. The queries are numbered 1 through 10. Below the queries, the execution results are shown as '1 row(s) inserted.' for each query. At the bottom, a status bar indicates the database is Oracle 19c EE Extreme Perf, running on a 2023 Oracle Linux 5.10 machine.

```
1 Insert Into Payments_History (reference_id, payment_id, loan_id, borrower_id, late_fees, balance_before_payment, balance_after_payment) values (5468754, 7558203, 3834054, 7202041, NULL, 30000.00, 180000.00);
2 Insert Into Payments_History (reference_id, payment_id, loan_id, borrower_id, late_fees, balance_before_payment, balance_after_payment) values (5577771, 6472888, 3104097, 6751480, NULL, 26000.00, 170000.00);
3 Insert Into Payments_History (reference_id, payment_id, loan_id, borrower_id, late_fees, balance_before_payment, balance_after_payment) values (5449579, 7207259, 3860302, 7721155, NULL, 26000.00, 170000.00);
4 Insert Into Payments_History (reference_id, payment_id, loan_id, borrower_id, late_fees, balance_before_payment, balance_after_payment) values (5565993, 6296799, 3421546, 7200864, NULL, 24000.00, 150000.00);
5 Insert Into Payments_History (reference_id, payment_id, loan_id, borrower_id, late_fees, balance_before_payment, balance_after_payment) values (5818379, 7015762, 2480497, 6187271, NULL, 30000.00, 150000.00);
6 Insert Into Payments_History (reference_id, payment_id, loan_id, borrower_id, late_fees, balance_before_payment, balance_after_payment) values (5842400, 7666370, 4050243, 6243070, NULL, 23000.00, 200000.00);
7 Insert Into Payments_History (reference_id, payment_id, loan_id, borrower_id, late_fees, balance_before_payment, balance_after_payment) values (5936167, 7013875, 3070004, 7538915, 50000.00, 22500.00, 80000.00);
8 Insert Into Payments_History (reference_id, payment_id, loan_id, borrower_id, late_fees, balance_before_payment, balance_after_payment) values (5280033, 7544981, 2255547, 6516977, NULL, 21500.00, NULL);
9 Insert Into Payments_History (reference_id, payment_id, loan_id, borrower_id, late_fees, balance_before_payment, balance_after_payment) values (5139199, 635400, 2029575, 6853965, NULL, 20000.00, NULL);
10 Insert Into Payments_History (reference_id, payment_id, loan_id, borrower_id, late_fees, balance_before_payment, balance_after_payment) values (5264361, 7127952, 4078770, 7857214, NULL, NULL, NULL);
```

1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.
1 row(s) inserted.

These SQL queries are inserting rows of data into a table called "Payments_History". The table has columns for "reference_id", "payment_id", "loan_id", "borrower_id", "late_fees", "balance_before_payment", and "balance_after_payment".

- Payments_History table: is used to track the payment history of loans in the system. Each row in the table represents a single payment made on a particular loan by a specific borrower.

- reference_id: column is a unique identifier for the payment record.
- payment_id: column is a unique identifier for the payment transaction.
- loan_id: column identifies the loan that the payment was made on.
- borrower_id: column identifies the borrower who made the payment.
- late_fees: column stores any fees that were incurred due to late payment.
- balance_before_payment: column stores the outstanding balance on the loan before the payment was made.
- balance_after_payment: column stores the outstanding balance on the loan after the payment was made.

Inserting rows of data into the "Payments_History" table allows the system to keep track of loan payments and generate reports on payment history and outstanding balances.

Query for Inserting data on payment history:

insert into Payments_History (reference_id, payment_id, loan_id, borrower_id, late_fees, balance_before_payment, balance_after_payment) values (5468754, 7358203, 3834054, 7200341, NULL, 30000.00, 180000.00);

insert into Payments_History (reference_id, payment_id, loan_id, borrower_id, late_fees, balance_before_payment, balance_after_payment) values (5577711, 6427088, 3104937, 6751403, NULL, 26000.00, 170000.00);

insert into Payments_History (reference_id, payment_id, loan_id, borrower_id, late_fees, balance_before_payment, balance_after_payment) values (5449579, 7267259, 3866302, 7721155, NULL, 25000.00, 170000.00);

insert into Payments_History (reference_id, payment_id, loan_id, borrower_id, late_fees, balance_before_payment, balance_after_payment) values (5366993, 6298799, 3431586, 7303864, NULL, 24000.00, 150000.00);

insert into Payments_History (reference_id, payment_id, loan_id, borrower_id, late_fees, balance_before_payment, balance_after_payment) values (5818379, 7015762, 2480497, 6187271, NULL, 23000.00, 150000.00);

insert into Payments_History (reference_id, payment_id, loan_id, borrower_id, late_fees, balance_before_payment, balance_after_payment) values (5842400, 7866370, 4050243, 6243070, NULL, 23000.00, 200000.00);

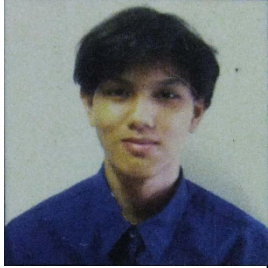


insert into Payments_History (reference_id, payment_id, loan_id, borrower_id, late_fees, balance_before_payment, balance_after_payment) values (5536167, 7013875, 3070934, 7558915, 50000.00, 22500.00, 80000.00);

insert into Payments_History (reference_id, payment_id, loan_id, borrower_id, late_fees, balance_before_payment, balance_after_payment) values (5280933, 7344981, 2255547, 6516977, NULL, 21500.00, NULL);

insert into Payments_History (reference_id, payment_id, loan_id, borrower_id, late_fees, balance_before_payment, balance_after_payment) values (5139199, 6325400, 2029575, 6853965, NULL, 20000.00, NULL);

insert into Payments_History (reference_id, payment_id, loan_id, borrower_id, late_fees, balance_before_payment, balance_after_payment) values (5264361, 7127952, 4078770, 7857214, NULL, NULL, NULL);

Summary of Assigned Tasks

Name (SN, FN, MI.)	Picture (1x1 formal attire)	Detailed Contributions/Assigned Tasks:
Leader's Name: Tiamzon, Bryan Dominick A.		Insertion of Data, Input values (every table), checking, documentation
Member 1: Faurillo, Ymwnl Jan		Created table (every table)
Member 2: Ureta, Juster S.		Documentation, Data input value (payments_history, payments)

Honor Pledge: "We affirm that we have not given or received any unauthorized help on this assignment, and that this work is our own"