

Case Study(Report)

‘Accounts Payable Database’

Student Details:

- *Rishabh Rajpurohit (9919103096)*
- *Soumya Gupta (9919103120)*
- *P. Paul Jonathan (9919103121)*



Database Systems and Web Lab

**Jaypee Institute of Information Technology
Noida**

July 11, 2021

ACKNOWLEDGEMENT

Presentation inspiration and motivation have always played a key role in the success of any venture.

We pay our deep sense of gratitude to, **Vice Chancellor Dr. S C Saxena, Jaypee institute of information technology, Noida** to encourage us to the highest peak and to provide us the opportunity to prepare the project. We are immensely obliged to for their elevating inspiration, encouraging guidance and kind supervision in the completion of our project.

We feel to acknowledge our indebtedness and deep sense of gratitude to our guide **Mrs. Vartika Puri** whose invaluable guidance and kind supervision given to us throughout the course which shaped the present work as its show.

Last, but not the least, **our parents** are also an important inspiration for time. So with due regards. We express our gratitude.

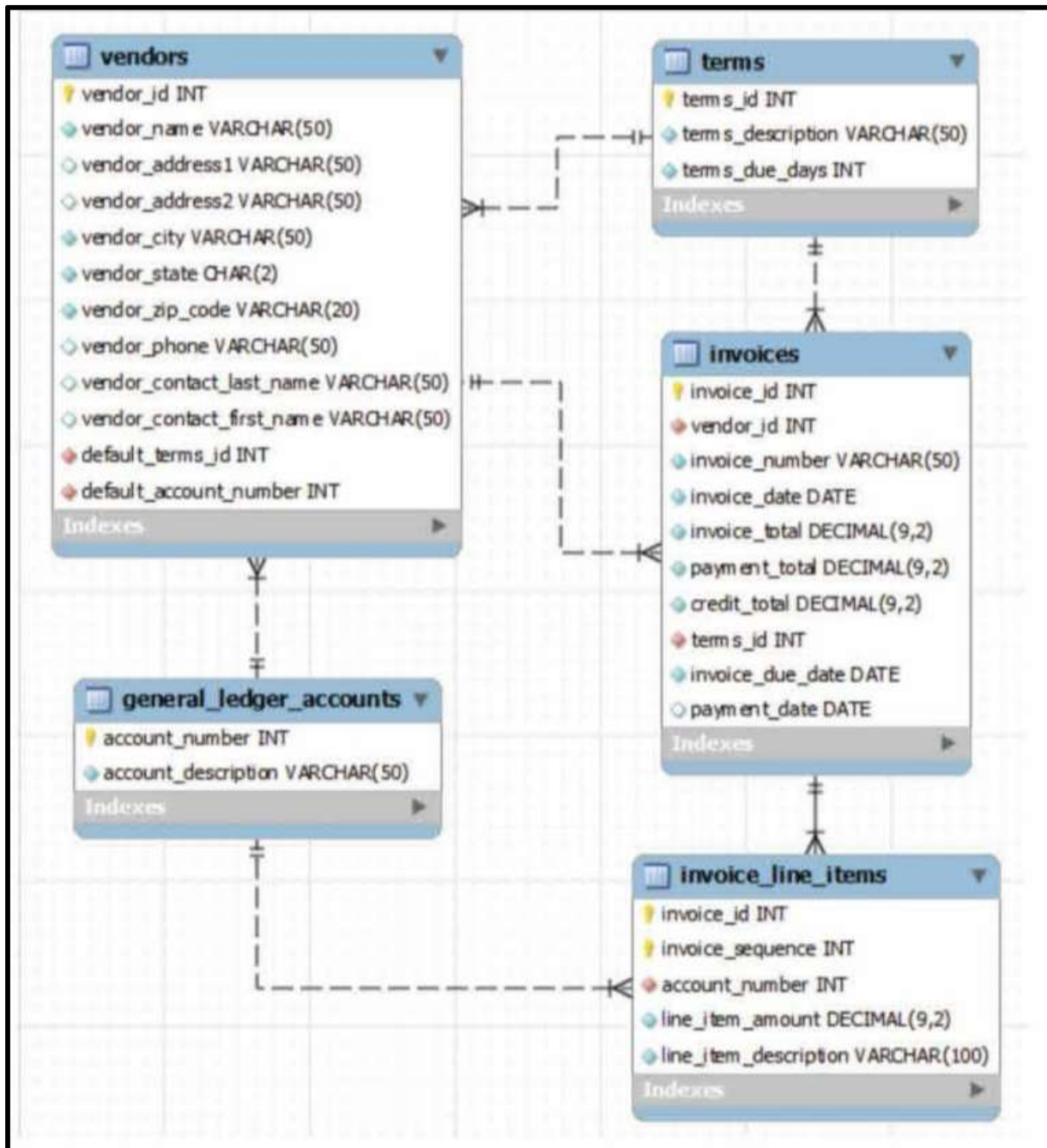
TABLE OF CONTENT

ACKNOWLEDGEMENT	2
1. PART:1	4
1.1 TITLE / EER	4
1.2 APPLICATIONS	5
2. PART:2	6
2.1 CREATING RELATIONS	6
2.2 INSERTING VALUES	7
3. PART:3	11
3.1 QUERY: TRIGGER 1	11
3.2 QUERY: TRIGGER 2	12
3.3 QUERY: TRIGGER 2	13
3.4 QUERY: PROCEDURE 1	14
3.5 QUERY: PROCEDURE 2	16
3.6 QUERY: PROCEDURE 3	20
3.7 QUERY: SCHEDULED EVENT 1	23
4. BIBLIOGRAPHY	24

1. PART:1

1.1 TITLE / EER

'Accounts Payable' Database



EER Diagram for the database

1.2 APPLICATIONS

By Implementing the above project in real time database:

- we can automate the manual stuff like keeping records of due payments
- sending alerts for upcoming payments
- charging penalties
- updating late fees daily automatically and
- offering discounts wherever needed

Thus this reduces the scope of error and greatly reduces human efforts.

2. PART:2

2.1 CREATING RELATIONS

```
use dswproject;
create table invoices(
  invoice_id int primary key AUTO_INCREMENT,
  vendor_id int not null,
  invoice_number VARCHAR(50) not null,
  invoice_date date not null,
  invoice_total decimal(9, 2) default 0,
  credit_total decimal(9, 2) DEFAULT 0,
  terms_id int not null,
  invoice_due_date date,
  payment_date date
);
CREATE table terms(
  terms_id INT PRIMARY key AUTO_INCREMENT,
  terms_description VARCHAR(50) NOT NULL,
  terms_due_days int
);
CREATE TABLE vendors(
  vendor_id int PRIMARY key AUTO_INCREMENT,
  vendor_name VARCHAR (50) not null,
  vendor_address1 VARCHAR (50) not null,
  vendor_address2 VARCHAR (50),
  vendor_city VARCHAR (50),
  vendor_state char(20),
  vendor_zip_code VARCHAR (20),
  vendor_phone VARCHAR (50),
  vendor_contact_last_name VARCHAR (50),
  vendor_contact_first_name VARCHAR (50),
  default_terms_id int not null,
  default_account_number int not null
);
CREATE table general_ledger_accounts(
  account_number int PRIMARY KEY AUTO_INCREMENT,
  account_description VARCHAR (50)
);
CREATE TABLE invoice_line_items(
  invoice_id int not null,
  invoice_sequence int not null,
  account_number int not null,
  line_item_amount decimal(9,2) not null,
  line_item_description VARCHAR (100),
  PRIMARY key(invoice_id, invoice_sequence),
  FOREIGN key(account_number) REFERENCES general_ledger_accounts (account_number)
);
alter table invoices
add constraint invoices_fk_vendors FOREIGN key (vendor_id) REFERENCES vendors (vendor_id),
add constraint invoices_fk_terms FOREIGN key (terms_id) REFERENCES terms (terms_id);
ALTER TABLE vendors
add constraint vendors_fk_terms FOREIGN key (default_terms_id) REFERENCES terms (terms_id),
add constraint vendors_fk_accounts FOREIGN key (default_account_number) REFERENCES general_ledger_accounts (account_number);
```

2.2 INSERTING VALUES

INSERT INTO

`general_ledger_accounts`

VALUES

(1, 'Illo quia ipsum itaque sit iste.')(2, 'Itaque eum et occaecati non repellendus.')(3, 'Sunt labore quia iure optio.')(4, 'Iusto dolor aut mollitia eius impedit asperiores.')(5, 'Voluptas dolorem enim ut omnis temporibus nulla.')

), -- 45 records more

INSERT INTO

`invoice_line_items`

VALUES

(103777,91708,1,986.41,'Distinctio et omnis impedit voluptatibus.')

),(109188,18779,24,6477.80,'Fuga aut consequatur ipsa sunt earum.')

),(114164,20835,34,1348369.18,'Sunt magnam veritatis architecto id dignissimos.')

),(138716,23728,35,0.00,'Architecto debitis autem libero aliquid sapiente.')

),(142781,76299,15,1165.00,'Non reiciendis molestias dolorum quaerat voluptatem consequatur sit.')

), --45 records more

INSERT INTO

`invoices` VALUES

(1,1,'43637','1991-04-24',1900.14,0.00,1,'1995-11-13','2006-02-04'),

(2,2,'450','1996-05-02',5899197.04,2047011.50,2,'1974-07-02','1971-06-08'),

(3,3,'4765','2003-08-05',78.71,9999999.99,3,'2006-12-23','2015-09-23'),

(4,4,'4627186','1998-10-06',3813.58,36568.31,4,'2007-03-13','2003-08-08'),

(5,5,'','1996-07-30',0.00,50261.92,5,'1992-10-23',

'2014-10-31'), --45 records more

```

INSERT INTO
`terms`
VALUES
(1, 'Quas qui aut occaecati.', 2019),(
2,
'Nihil modi omnis dolorum voluptate eligendi repell',
1972
),
3,
'Sed qui rerum consequatur sit reprehenderit modi.',
2008
),(4, 'Sint maiores consequatur sed.', 1989),(
5,
'Cumque illum et qui iusto praesentium sit enim.',
2007
), --45 records more

```

```

INSERT INTO
`vendors`
VALUES
(1,'Carter Jerde MD','50300 Rosanna Point Suite 746','74703 Rohan Views','Cassandrehaven','Wisconsin','460878','07965412824','139.711.9651','01868 411465',1,1),
(2,'Prof. Judd Harris','0745 Streich Court Suite 482','08065 Graham Court Suite 773','Moriahville','Indiana','411798','1-279-686-9444','+41(3)2483486604','(190)302 -9905x54471',2,2),
(3,'Dandre Koepp','9256 Lockman Corner Suite 353','535 Mortimer Haven Suite 537','Lake Estevan','Vermont','09986359834','832026','392-932-6974x475','336.615.1911x199',3,3),
(4,'Dan Lesch','4378 Corkery Lock Suite 432','64436 Simonis Lights','California','West Anastasiaberg','133-511-4358','883743','852-915-4148x546','1-797-074-8152',4,4),
(5,'20214 Aidan Highway','Ms. Antonietta Brown IV','Lake Annamarietvill','453 Sipes Mills Apt. 200','387820','Pennsylvania','864.316.0417','(138)284-0952x3786','1-497-025-9384',5,5), --45 records more

```

```
mysql> select * from general_ledger_accounts
-> where account_number IN (1,2,3,4,5);
```

account_number	account_description
1	Illo quia ipsum itaque sit iste.
2	Itaque eum et occaecati non repellendus.
3	Sunt labore quia iure optio.
4	Iusto dolor aut mollitia eius impedit asperiores.
5	Voluptas dolorem enim ut omnis temporibus nulla.

```
5 rows in set (0.00 sec)
```

```
mysql> select * from terms
-> where terms_id IN (1,2,3,4,5);
```

terms_id	terms_description	terms_due_days
1	Quas qui aut occaecati.	2019
2	Nihil modi omnis dolorum voluptate eligendi repell	1972
3	Sed qui rerum consequatur sit reprehenderit modi.	2008
4	Sint maiores consequatur sed.	1989
5	Cumque illum et qui iusto praesentium sit enim.	2007

```
5 rows in set (0.00 sec)
```

```
mysql>
mysql> select invoice_id,invoice_sequence,account_number,line_item_amount from invoice_line_items
-> where account_number IN (1,24,34,50);
```

invoice_id	invoice_sequence	account_number	line_item_amount
103777	91708	1	986.41
109188	18779	24	6477.80
114164	20835	34	1348369.18
845537	30361	50	1.56

```
4 rows in set (0.00 sec)
```

```
mysql> select invoice_id,invoice_date,invoice_total from invoices
-> where invoice_id IN (1,2,3,4,5);
```

invoice_id	invoice_date	invoice_total
1	1991-04-24	39153.00
2	1996-05-02	100000.00
3	2003-08-05	1633.00
4	1998-10-06	3813.58
5	1996-07-30	0.00

```
5 rows in set (0.00 sec)
```

```
mysql>
mysql> select vendor_id,vendor_name from vendors
-> where vendor_id IN (1,2,3,4,5);
+-----+-----+
| vendor_id | vendor_name |
+-----+-----+
| 1 | Carter Jerde MD |
| 2 | Prof. Judd Harris |
| 3 | Dandre Koepp |
| 4 | Dan Lesch |
| 5 | Ms. Antonietta Brown IV |
+-----+-----+
5 rows in set (0.00 sec)
```

3. PART:3

3.1 QUERY: TRIGGER 1

```
/*1 trigger: BEFORE delete on terms table as it is foreign key constrained*/
DELIMITER //
CREATE TRIGGER deleteTermsNotAllowed
BEFORE DELETE ON terms FOR EACH ROW
BEGIN
signal sqlstate '49001' SET message_text = "DELETE on terms forbidden (foreign key restrained)!" ;
END //
DELIMITER ;
DELETE FROM terms;
```

```
mysql> CREATE TRIGGER deleteTermsNotAllowed
-> BEFORE DELETE ON terms FOR EACH ROW
-> BEGIN
-> signal sqlstate '49001' set message_text ="DELETE on terms forbidden (foreign key restrained)!" ;
-> END //
Query OK, 0 rows affected (0.01 sec)

mysql> delimiter ;
mysql>
mysql> delete from terms
-> ;
ERROR 1644 (49001): DELETE on terms forbidden (foreign key restrained)!
```

3.2 QUERY: TRIGGER 2

```
/*2 trigger: BEFORE delete on general Ledger Accounts table as it is foreign key constrained*/  
DELIMITER //  
CREATE TRIGGER deleteAccountsNotAllowed  
BEFORE DELETE ON general_ledger_accounts FOR EACH ROW  
BEGIN  
signal sqlstate '49002' SET message_text = "DELETE on general_ledger_accounts forbidden (foreign key restrained)!";  
END //  
DELIMITER ;  
DELETE FROM general_ledger_accounts;
```

```
mysql>  
mysql> delimiter //  
mysql> CREATE TRIGGER deleteAccountNotAllowed  
-> BEFORE DELETE ON general_ledger_accounts FOR EACH ROW  
-> BEGIN  
-> signal sqlstate '49002' set message_text ="DELETE on GEN. LEDGER ACCOUNTS forbidden (foreign key restrained)!";  
-> END //  
Query OK, 0 rows affected (0.01 sec)  
  
mysql> delimiter ;  
mysql> delete * from general_ledger_accounts;  
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '* from general_ledger_accounts' at line 1  
mysql> delete from general_ledger_accounts;  
ERROR 1644 (49002): DELETE on GEN. LEDGER ACCOUNTS forbidden (foreign key restrained)!  
mysql>
```


3.3 QUERY: TRIGGER 2

```
/*3 trigger: BEFORE INSERT on invoices table discount 10% if invoice_total > 100000
*/
DELIMITER //
CREATE TRIGGER updateInvoiceDiscount
BEFORE INSERT ON invoices FOR EACH ROW
BEGIN
IF new.invoice_total > 100000 THEN
SET new.invoice_total = (new.invoice_total)*0.9;
END IF;
END //
DELIMITER ;
INSERT INTO invoices(vendor_id,invoice_number,invoice_date,invoice_total,credit_tot
al,terms_id,invoice_due_date,payment_date)
VALUES (50,'499700','2012-08-23',1000099,48379.32,1,'2012-11-23','2012-09-23');
SELECT (invoice_total) FROM invoices
WHERE vendor_id = 50 AND terms_id = 1;
```

```
mysql> CREATE TRIGGER updateInvoiceDiscount
-> BEFORE INSERT ON invoices FOR EACH ROW
-> BEGIN
-> IF new.invoice_total > 100000 THEN
-> SET new.invoice_total = (new.invoice_total)*0.9;
-> END IF;
-> END //
Query OK, 0 rows affected (0.01 sec)

mysql> delimiter ;
```

```
mysql> insert into invoices(vendor_id,invoice_number,invoice_date,invoice_total,credit_total,
terms_id,invoice_due_date,payment_date) values
-> (50,'499700','2012-08-23',1000099,48379.32,1,'2012-11-23','2012-09-23');
Query OK, 1 row affected (0.00 sec)

mysql> SELECT (invoice_total) from invoices
-> where vendor_id = 50 AND terms_id = 1;
+-----+
| invoice_total |
+-----+
|      10000.99 |
|    900089.10 |
|    900089.10 |
+-----+
3 rows in set (0.00 sec)

mysql> |
```

3.4 QUERY: PROCEDURE 1

```
/*4 Procedure: To get names of all vendors*/
DELIMITER //
CREATE PROCEDURE getvendors(INOUT vendorslist VARCHAR (10000))
BEGIN
DECLARE isdone INT DEFAULT 0;
DECLARE vendorname varchar(50);
DECLARE vendorcursor CURSOR FOR
SELECT vendor_name from vendors;
DECLARE CONTINUE HANDLER FOR NOT FOUND SET isdone = 1;
OPEN vendorcursor;
vendorloop: LOOP
FETCH vendorcursor INTO vendorname;
IF isdone = 1 THEN
LEAVE vendorloop;
END IF;
SET vendorslist = CONCAT(vendorname, ", ", vendorslist);
END LOOP vendorloop;
CLOSE vendorcursor;
END //

DELIMITER ;
SET @vendorlist = "";
CALL getvendors(@vendorlist);
SELECT @vendorlist;
```

```
mysql> delimiter //
mysql> CREATE PROCEDURE getvendors(INOUT vendorslist VARCHAR (10000))
-> BEGIN
-> DECLARE isdone INT DEFAULT 0;
-> DECLARE vendorname varchar(50);
-> DECLARE vendorcursor CURSOR FOR
-> SELECT vendor_name from vendors;
-> DECLARE CONTINUE HANDLER FOR NOT FOUND SET isdone = 1;
-> OPEN vendorcursor;
-> vendorloop: LOOP
-> FETCH vendorcursor INTO vendorname;
-> IF isdone = 1 THEN
-> LEAVE vendorloop;
-> END IF;
-> SET vendorslist = CONCAT(vendorname, ", ", vendorslist);
-> END LOOP vendorloop;
-> CLOSE vendorcursor;
-> END //
Query OK, 0 rows affected (0.01 sec)

mysql> delimiter ;
mysql> set @vendorlist = "";
Query OK, 0 rows affected (0.00 sec)

mysql> call getvendors(@vendorlist);
Query OK, 0 rows affected (0.01 sec)

mysql> select @vendorlist;
```

OUTPUT:

| Mertie Spinka, Constantin Schneider, Miss Martina Grant MD, Koby Fahey, Dr. Isaias Sauer, Tyshawn Hermiston, Monica Gerhold, Josianne Luetngen, Gunner Abbott, Isac Rosenbaum, Santana Schultz, Gregg Nader, Torey Fadel, Mr. Walton Bosco, Frederick Morar, Dr. Ethelyn Hoeger DVM, Dr. Colleen Purdy, Claudie DuBuque, Agustin Cummerata, Dr. Americo Schoen Jr., Thea Gaylord, Aaron Friesen, Henriette Cassin, Jarrod Gleason MD, Dr. Garnet Gorczany IV, Fredrick Ferry, Arlo DuBuque, Lon Waters, Haylie Kutch, Mikayla Stark, Idella Rempel, Mrs. Leta Daugherty II, Jovan Gerlach, Berniece Blanda, Sonny Hilll, Ocie Schowalter, Abdullah Greenfelder, Marisol Ankunding, Beau Will, Haylee Homenick, Dion DuBuque, Matt Reinger, Prof. Jeramie Jacobs V, Aliya Gerhold, Mrs. Lillian Rohan, Ms. Antonietta Brown IV, Dan Lesch, Dandre Koepp, Prof. Judd Harris, Carter Jerde MD, |

3.5 QUERY: PROCEDURE 2

```
/*5: procedure to add late fees*/  
-- there were no dates greater than current date so added a few  
update invoices  
SET payment_date = '2021-10-21'  
WHERE invoice_id IN (1,3,5,7,9);  
  
-- some invoice_total values were offlimit, thus  
update invoices  
SET invoice_total = 100000  
WHERE invoice_total > 100000  
-- glimpse of values about to be updated  
select invoice_id,invoice_total,'NOT updated payment' from invoices  
WHERE invoice_id IN (1,3,5,7,9);
```

```
mysql> update invoices  
-> SET payment_date = '2021-10-21'  
-> WHERE invoice_id IN (1,3,5,7,9);  
Query OK, 5 rows affected (0.00 sec)  
Rows matched: 5  Changed: 5  Warnings: 0
```

```
mysql> update invoices  
-> SET invoice_total = 100000  
-> WHERE invoice_total > 100000  
-> ;  
Query OK, 19 rows affected (0.00 sec)  
Rows matched: 19  Changed: 19  Warnings: 0
```

invoice_id	invoice_total	NOT updated payment
1	2095.00	NOT updated payment
3	87.00	NOT updated payment
5	0.00	NOT updated payment
7	100000.00	NOT updated payment
9	100000.00	NOT updated payment


```

-- procedure
DELIMITER //
CREATE procedure addlatepayment()
BEGIN
DECLARE isdone INT DEFAULT 0;
DECLARE invid INT DEFAULT 0;
DECLARE invtotal INT DEFAULT 0;
DECLARE oldpayment INT DEFAULT 0;
DECLARE invpdate date DEFAULT 0;
DECLARE invcursor CURSOR FOR
SELECT invoice_id,payment_date,invoice_total FROM invoices;
DECLARE CONTINUE HANDLER FOR NOT FOUND SET isdone = 1;
OPEN invcursor;
invloop: LOOP
FETCH invcursor INTO invid,invpdate,invtotal;
SET oldpayment = invtotal;
IF isdone = 1 THEN
LEAVE invloop;
END IF;
IF invpdate > CURDATE() THEN
SET invtotal = invtotal * 1.05;
UPDATE invoices
SET invoice_total = invtotal
WHERE invoice_id = invid;
SELECT invoice_id as ID,oldpayment as PREVIOUS_TOTAL,invoice_total as UPDATED_TOTAL,
'UPDATED' as STATUS from invoices
WHERE invoice_id = invid;
END IF;
END LOOP invloop;
SELECT 'Late dues added successfully';
CLOSE invcursor;
END //
DELIMITER ;

-- calling PROCEDURE
call addlatepayment();
--actually whole table is traversed but
--here only those tuples shown which were actually updated

```

```

mysql> delimiter //
mysql> CREATE procedure addlatepayment()
-> BEGIN
-> DECLARE isdone INT DEFAULT 0;
-> DECLARE invid INT DEFAULT 0;
-> DECLARE invtotal INT DEFAULT 0;
-> DECLARE oldpayment INT DEFAULT 0;
-> DECLARE invpdate date DEFAULT 0;
-> DECLARE invcursor CURSOR FOR
-> SELECT invoice_id,payment_date,invoice_total FROM invoices;
-> DECLARE CONTINUE HANDLER FOR NOT FOUND SET isdone = 1;
-> OPEN invcursor;
-> invloop: LOOP
-> FETCH invcursor INTO invid,invpdate,invtotal;
-> SET oldpayment = invtotal;
-> IF isdone = 1 THEN
-> LEAVE invloop;
-> END IF;
-> IF invpdate > CURDATE() THEN
-> SET invtotal = invtotal * 1.05;
-> UPDATE invoices
-> SET invoice_total = invtotal
-> WHERE invoice_id = invid;
-> SELECT invoice_id as ID,oldpayment as PREVIOUS_TOTAL,invoice_total
as UPDATED_TOTAL,'UPDATED' as STATUS from invoices
-> WHERE invoice_id = invid;
-> END IF;
-> END LOOP invloop;
-> SELECT 'Late dues added successfully';
-> CLOSE invcursor;
-> END //
Query OK, 0 rows affected (0.00 sec)

mysql> delimiter ;

```

```
mysql> call addlatepayment();
```

```
+-----+-----+-----+-----+
| ID | PREVIOUS_TOTAL | UPDATED_TOTAL | STATUS |
+-----+-----+-----+-----+
| 1 | 2095 | 2200.00 | UPDATED |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

```
+-----+-----+-----+-----+
| ID | PREVIOUS_TOTAL | UPDATED_TOTAL | STATUS |
+-----+-----+-----+-----+
| 3 | 87 | 91.00 | UPDATED |
+-----+-----+-----+-----+
1 row in set (0.02 sec)
```

```
+-----+-----+-----+-----+
| ID | PREVIOUS_TOTAL | UPDATED_TOTAL | STATUS |
+-----+-----+-----+-----+
| 5 | 0 | 0.00 | UPDATED |
+-----+-----+-----+-----+
1 row in set (0.02 sec)
```

```
+-----+-----+-----+-----+
| ID | PREVIOUS_TOTAL | UPDATED_TOTAL | STATUS |
+-----+-----+-----+-----+
| 7 | 100000 | 105000.00 | UPDATED |
+-----+-----+-----+-----+
1 row in set (0.03 sec)
```

```
+-----+-----+-----+-----+
| ID | PREVIOUS_TOTAL | UPDATED_TOTAL | STATUS |
+-----+-----+-----+-----+
| 9 | 100000 | 105000.00 | UPDATED |
+-----+-----+-----+-----+
1 row in set (0.04 sec)
```

```
+-----+
| Late dues added successfully |
+-----+
| Late dues added successfully |
+-----+
1 row in set (0.05 sec)
```

```
Query OK, 0 rows affected (0.05 sec)
```

3.6 QUERY: PROCEDURE 3

```
/*6: procedure to show due payments*/

-- procedure
DELIMITER //
CREATE procedure showduepayments()
BEGIN
DECLARE isdone INT DEFAULT 0;
DECLARE invid INT DEFAULT 0;
DECLARE invtotal INT DEFAULT 0;
DECLARE invpdate date DEFAULT 0;
DECLARE cursorinv CURSOR FOR
SELECT invoice_id,payment_date,invoice_total FROM invoices;
DECLARE CONTINUE HANDLER FOR NOT FOUND SET isdone = 1;
OPEN cursorinv;
invloop: LOOP
FETCH cursorinv INTO invid,invpdate,invtotal;
IF isdone = 1 THEN
LEAVE invloop;
END IF;
IF invpdate > CURDATE() THEN
SELECT invoice_id as ID,payment_date,invoice_total as DUE_INVOICE,'PENDING' as STATUS from invoices
WHERE invoice_id = invid;
END IF;
END LOOP invloop;
CLOSE cursorinv;
END //
DELIMITER ;
-- calling PROCEDURE
call showduepayments();
```



```

mysql> delimiter //
mysql> CREATE procedure showduepayments()
-> BEGIN
-> DECLARE isdone INT DEFAULT 0;
-> DECLARE invid INT DEFAULT 0;
-> DECLARE invtotal INT DEFAULT 0;
-> DECLARE invpdate date DEFAULT 0;
-> DECLARE cursorinv CURSOR FOR
-> SELECT invoice_id,payment_date,invoice_total FROM invoices;
-> DECLARE CONTINUE HANDLER FOR NOT FOUND SET isdone = 1;
-> OPEN cursorinv;
-> invloop: LOOP
-> FETCH cursorinv INTO invid,invpdate,invtotal;
-> IF isdone = 1 THEN
-> LEAVE invloop;
-> END IF;
-> IF invpdate > CURDATE() THEN
-> SELECT invoice_id as ID,payment_date,invoice_total as DUE_INVOICE,
'PENDING' as STATUS from invoices
-> WHERE invoice_id = invid;
-> END IF;
-> END LOOP invloop;
-> CLOSE cursorinv;
-> END //
Query OK, 0 rows affected (0.00 sec)

mysql> delimiter ;

```

```
mysql> call showduepayments;
```

ID	payment_date	DUE_INVOICE	STATUS
1	2021-10-21	2200.00	PENDING

1 row in set (0.00 sec)

ID	payment_date	DUE_INVOICE	STATUS
3	2021-10-21	91.00	PENDING

1 row in set (0.01 sec)

ID	payment_date	DUE_INVOICE	STATUS
5	2021-10-21	0.00	PENDING

1 row in set (0.02 sec)

ID	payment_date	DUE_INVOICE	STATUS
7	2021-10-21	105000.00	PENDING

1 row in set (0.03 sec)

ID	payment_date	DUE_INVOICE	STATUS
9	2021-10-21	105000.00	PENDING

1 row in set (0.04 sec)

Query OK, 0 rows affected (0.05 sec)

3.7 QUERY: SCHEDULED EVENT 1

```
/*7: Timestamp trigger to run procedure addlatepayment() daily once*/
-- to run scheduled events event organiser should be ON
-- set it to 0 to turn it OFF
SET GLOBAL event_scheduler = 1;
-- sample event to see results if working
CREATE EVENT updatepayment
ON SCHEDULE EVERY 5 SECOND
DO
CALL dswproject.addlatepayment();
select invoice_id,invoice_total from invoices
where invoice_id IN (1,3,5,7,9);
-- repeat after 5 seconds
select invoice_id,invoice_total from invoices
where invoice_id IN (1,3,5,7,9);
--dropping sample event
drop event if exists updatepayment;
```

```
mysql> SET GLOBAL event_scheduler = 1;
Query OK, 0 rows affected (0.00 sec)

mysql> CREATE EVENT updatepayment
-> ON SCHEDULE EVERY 5 SECOND
-> DO
-> CALL dswproject.addlatepayment();
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> select invoice_id,invoice_total from invoices
-> where invoice_id IN (1,3,5,7,9);
+-----+-----+
| invoice_id | invoice_total |
+-----+-----+
| 1 | 24036.00 |
| 3 | 1002.00 |
| 5 | 0.00 |
| 7 | 1146763.00 |
| 9 | 1146763.00 |
+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> select invoice_id,invoice_total from invoices
-> where invoice_id IN (1,3,5,7,9);
+-----+-----+
| invoice_id | invoice_total |
+-----+-----+
| 1 | 30677.00 |
| 3 | 1279.00 |
| 5 | 0.00 |
| 7 | 1463592.00 |
| 9 | 1463592.00 |
+-----+-----+
5 rows in set (0.00 sec)
```

4. BIBLIOGRAPHY

- [1] MySQL scheduled events
<https://www.mysqltutorial.org/mysql-triggers/working-mysql-schedule-event/>
- [2] MySQL Reference Manual
<https://dev.mysql.com/doc/>
- [3] Murach's MySQL 3rd Edition (book)
- [4] StackOverflow (for finding solutions to errors)

END