
LABO 4 BDR

-- Exercice 01

```
drop trigger if exists payment_date;
DROP TRIGGER IF EXISTS trigEx1;
DELIMITER $$
create trigger trigEx1
before insert on payment
for each row
begin
    set
        NEW.amount = NEW.amount * 1.08,
        NEW.last_update = NOW();
end; $$
```

DELIMITER ;

-- END Exercice 01**-- Exercice 01_insert**

```
insert into payment(customer_id, staff_id, rental_id, amount, payment_date)
values (1, 1, 1, 1, now());
```

-- END Exercice 01_insert**-- Exercice 01_affichage**

```
select
    customer_id,
    staff_id,
    rental_id,
    amount,
    payment_date
from payment
where rental_id = 1;
```

-- END Exercice 01_affichage

customer_id	staff_id	rental_id	amount	payment_date
130	1	1	2.99	2005-05-24 22:53:30

customer_id	staff_id	rental_id	amount	payment_date
130	1	1	2.99	2005-05-24 22:53:30
1	1	1	1.08	2015-12-18 17:00:41

-- Exercice 02_a

```
CREATE TABLE staff_creation_log (
    username varchar(16) NOT NULL,
    when_created TIMESTAMP NOT NULL
)ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

-- END Exercice 02_a**-- Exercice 02_b**

```
DROP TRIGGER IF EXISTS trigEx2;
DELIMITER $$
```

```
create trigger trigEx2
after insert on staff
for each row
begin
    insert into staff_creation_log (username, when_created)
    values (
        NEW.username,
        NEW.last_update
    );
end; $$
```

DELIMITER ;

-- END Exercice 02_b

-- Exercice 02_insert

```
insert into staff(first_name, last_name, address_id, email, store_id, active, username, last_update)
values ("Jean", "Dupond", 5, "jean.dupond@gmail.com", 3, 1, "JDup", NOW());
```

-- END Exercice 02_insert

-- Exercice 02_affichage

```
select *
from staff_creation_log
where username = "JDup";
```

-- END Exercice 02_affichage

username	when_created
JDup	2015-12-18 17:01:21

-- Exercice 03

```
DROP TRIGGER IF EXISTS trigEx3;
```

DELIMITER \$\$

```
create trigger trigEx3
```

```
before insert on staff
```

```
for each row
```

```
begin
```

```
    set
```

```
        NEW.email = concat(NEW.first_name, '.', New.last_name, '@sakilastaff.com');
```

```
end; $$
```

DELIMITER ;

-- END Exercice 03

-- Exercice 03_insert

```
insert into staff(first_name, last_name, address_id, email, store_id, active, username, last_update)
values ("Jean", "Dupond", 5, "jean.dupond@gmail.com", 3, 1, "JDup", NOW());
```

-- END Exercice 03_insert

-- Exercice 03_affichage

```
select
```

```
    first_name,
```

```
    last_name,
```

```
    email
```

```
from staff
```

```
where username = "JDUP";
```

-- END Exercice 03_affichage

first_name	last_name	email
Jean	Dupond	jean.dupond@gmail.com

first_name	last_name	email
Jean	Dupond	jean.dupond@gmail.com
Jean	Dupond	Jean.Dupond@sakilastaff.com

-- Exercise 04_a

```
DROP TABLE IF EXISTS customer_store_log;
create table customer_store_log(
```

```
    customer_id int,
    last_store_id int,
    register_date DATE,
    unregister_date DATE
```

```
)ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

-- END Exercise 04_a**-- Exercise 04_b**

```
DROP TRIGGER IF EXISTS trigEx4;
```

```
DELIMITER $$
```

```
create trigger trigEx4
```

```
before update on customer
```

```
for each row
```

```
begin
```

```
    if OLD.store_id <> NEW.store_id
```

```
    then
```

```
        insert into customer_store_log(customer_id, last_store_id, register_date, unregister_date)
        values(OLD.customer_id, OLD.store_id, OLD.last_update, now());
```

```
    end if;
```

```
end;$$
```

```
DELIMITER ;
```

-- END Exercise 04_b**-- Exercise 04_update**

```
update customer set store_id = 4 where customer_id = 1;
```

-- END Exercise 04_update**-- Exercise 04_affichage**

```
select *
```

```
from customer_store_log
```

```
where customer_id = 1;
```

-- END Exercise 04_affichage

customer_id	last_store_id	register_date	unregister_date
1	1	2006-02-15	2015-12-18

-- Exercise 05

```
set global event_scheduler = ON;
```

```
drop event if exists delete_row;
```

```
create event delete_row on schedule every 1 minute
```

```
on completion preserve
```

```
do
```

```
    delete from customer_store_log
    where customer_id in (
```

```

select
    customer_id
from (
    select
        customer_id
    from customer_store_log
    group by customer_id
    having
        count(customer_id) > 1
    and
        max(unregister_date) < now() - interval 1 year
    ) as tab1
);
-- END Exercise 05

```

-- Exercise 06_a

```

drop view if exists vue_ex6;
create view vue_ex6
as
select
    phone,
    address
from staff
inner join address
on staff.address_id = address.address_id;

```

-- END Exercise 06_a

-- Exercise 06_a_affichage

```

select *
from vue_ex6
limit 20;
-- END Exercise 06_a_affichage

```

phone	address
14033335568	23 Workhaven Lane
6172235589	1411 Lillydale Drive
28303384290	1913 Hanoi Way
28303384290	1913 Hanoi Way

-- Exercise 06_b

Oui Franklin pourra mettre à jour la base de donnée à travers cette vue car elle ne contient aucune des catégories bloquant les vues updatables :

pas de fonction d'aggregation, ni de distinct, ni de group by, ni de having, union, sous-requête dans un select, sous-requête dans un where, etc..

```
update vue_ex6 set phone = '4' where phone = '14033335568';
```

-- END Exercise 06_b

-- Exercise 06_b_affichage

```

select
    address,
    phone
from address
where address_id = 3;

```

-- END Exercice 06_b_affichage

address	phone
23 Workhaven Lane	14033335568

address	phone
23 Workhaven Lane	4

-- Exercice 07

drop view if exists vue_ex7;

create view vue_ex7

as

```

select
    email,
    film.title as 'titre',
    datediff(NOW(), rental.rental_date + interval film.rental_duration day) as 'jours_retard'
from rental
inner join customer
    on rental.customer_id = customer.customer_id
inner join inventory
    on rental.inventory_id = inventory.inventory_id
inner join film
    on inventory.film_id = film.film_id
where rental.return_date is null
having jours_retard > 0;
```

-- END Exercice 07**-- Exercice 07_affichage**

select *

from vue_ex7

limit 20;

-- END Exercice 07_affichage

email	titre	jours_retard
ELIZABETH.BROWN@sakilacustomer.org	GABLES METROPOLIS	3591
MARGARET.MOORE@sakilacustomer.org	DEEP CRUSADE	3588
LISA.ANDERSON@sakilacustomer.org	BULL SHAWSHANK	3588
BETTY.WHITE@sakilacustomer.org	GROUNDHOG UNCUT	3588
HELEN.HARRIS@sakilacustomer.org	CANDIDATE PERDITION	3590
HELEN.HARRIS@sakilacustomer.org	SMOKING BARBARELLA	3587
MICHELLE.CLARK@sakilacustomer.org	DWARFS ALTER	3588
LAURA.RODRIGUEZ@sakilacustomer.org	SUIT WALLS	3591
SARAH.LEWIS@sakilacustomer.org	DIVINE RESURRECTION	3590
CYNTHIA.YOUNG@sakilacustomer.org	SUSPECTS QUILLS	3590
ANGELA.HERNANDEZ@sakilacustomer.org	CANYON STOCK	3587
ANNA.HILL@sakilacustomer.org	BOULEVARD MOB	3591
STEPHANIE.MITCHELL@sakilacustomer....	SHOCK CABIN	3587
CAROLYN.PEREZ@sakilacustomer.org	TORQUE BOUND	3591
CAROLYN.PEREZ@sakilacustomer.org	RIVER OUTLAW	3590
CHRISTINE.ROBERTS@sakilacustomer.org	POLLOCK DELIVERANCE	3589
CHRISTINE.ROBERTS@sakilacustomer.org	MOTIONS DETAILS	3589
MARIE.TURNER@sakilacustomer.org	AMERICAN CIRCUS	3591
JULIE.SANCHEZ@sakilacustomer.org	TRADING PINOCCHIO	3588
HEATHER.MORRIS@sakilacustomer.org	PEACH INNOCENT	3591

-- Exercise 08

```
select *
from vue_ex7
where jours_retard > 3
limit 20;
```

-- END Exercise 08

email	titre	jours_retard
ELIZABETH.BROWN@sakilacustomer.org	GABLES METROPOLIS	3591
MARGARET.MOORE@sakilacustomer.org	DEEP CRUSADE	3588
LISA.ANDERSON@sakilacustomer.org	BULL SHAWSHANK	3588
BETTY.WHITE@sakilacustomer.org	GROUNDHOG UN CUT	3588
HELEN.HARRIS@sakilacustomer.org	CANDIDATE PERDITION	3590
HELEN.HARRIS@sakilacustomer.org	SMOKING BARBARELLA	3587
MICHELLE.CLARK@sakilacustomer.org	DWARFS ALTER	3588
LAURA.RODRIGUEZ@sakilacustomer.org	SUIT WALLS	3591
SARAH.LEWIS@sakilacustomer.org	DIVINE RESURRECTION	3590
CYNTHIA.YOUNG@sakilacustomer.org	SUSPECTS QUILLS	3590
ANGELA.HERNANDEZ@sakilacustomer.org	CANYON STOCK	3587
ANNA.HILL@sakilacustomer.org	BOULEVARD MOB	3591
STEPHANIE.MITCHELL@sakilacustomer.org	SHOCK CABIN	3587
CAROLYN.PEREZ@sakilacustomer.org	TORQUE BOUND	3591
CAROLYN.PEREZ@sakilacustomer.org	RIVER OUTLAW	3590
CHRISTINE.ROBERTS@sakilacustomer.org	POLLOCK DELIVERANCE	3589
CHRISTINE.ROBERTS@sakilacustomer.org	MOTIONS DETAILS	3589
MARIE.TURNER@sakilacustomer.org	AMERICAN CIRCUS	3591
JULIE.SANCHEZ@sakilacustomer.org	TRADING PINOCCHIO	3588
HEATHER.MORRIS@sakilacustomer.org	PEACH INNOCENT	3591

-- Exercise 09_a

```
drop view if exists vue_ex9;
create view vue_ex9
as
select
    customer.first_name as 'prénom',
    customer.last_name as 'nom',
    customer.customer_id as 'id',
    count(*) as 'nombre_locations'
from rental
inner join customer
    on rental.customer_id = customer.customer_id
group by customer.customer_id;
```

-- END Exercise 09_a**-- Exercise 09_a_affichage**

```
select *
from vue_ex9
limit 20;
```

-- END Exercise 09_a_affichage

prénom	nom	id	nombre_locations
MARY	SMITH	1	47
PATRICIA	JOHNSON	2	27
LINDA	WILLIAMS	3	26
BARBARA	JONES	4	22
ELIZABETH	BROWN	5	38
JENNIFER	DAVIS	6	28
MARIA	MILLER	7	33
SUSAN	WILSON	8	24
MARGARET	MOORE	9	23
DOROTHY	TAYLOR	10	25
LISA	ANDERSON	11	24
NANCY	THOMAS	12	28
KAREN	JACKSON	13	27
BETTY	WHITE	14	28
HELEN	HARRIS	15	32
SANDRA	MARTIN	16	28
DONNA	THOMPSON	17	21
CAROL	GARCIA	18	22
RUTH	MARTINEZ	19	24
SHARON	ROBINSON	20	30

-- Exercice 09_b

select

prénom,

nom,

id,

max(nombre_locations) as nombre_locations

from vue_ex9

limit 20;

-- END Exercice 09_b

prénom	nom	id	nombre_locations
MARY	SMITH	1	47

-- Exercice 10_a

drop view if exists vue_ex10;

create view vue_ex10

as

select

DATE(rental_date) as jour_location,

count(*) as nombre

from rental

group by jour_location;

-- END Exercice 10_a**-- Exercice 10_a_affichage**

select *

from vue_ex10

limit 20;

-- END Exercice 10_a_affichage

jour_location	nombre
2005-05-24	8
2005-05-25	137
2005-05-26	174
2005-05-27	166
2005-05-28	196
2005-05-29	154
2005-05-30	158
2005-05-31	163
2005-06-14	16
2005-06-15	348
2005-06-16	324
2005-06-17	325
2005-06-18	344
2005-06-19	348
2005-06-20	331
2005-06-21	275
2005-07-05	27
2005-07-06	504
2005-07-07	461
2005-07-08	512

-- Exercise 10_b

```
select *
from vue_ex10
where jour_location = "2005-08-01";
```

-- END Exercise 10_b

jour_location	nombre
2005-08-01	671

-- Exercise 11_a

```
drop function if exists func_ex11;
delimiter //
create function func_ex11 (id_store int)
returns int
begin
declare nombre int;
select
    count(*)
into nombre
from (
    select distinct
        film_id
    from
        inventory
    where
        store_id = id_store
    ) as a;
return nombre;
end; //
```

```
delimiter ;
```

-- END Exercise 11_a

-- Exercice 11_b

select

```
func_ex11(1) as nombre_films_store_1,
func_ex11(2) as nombre_films_store_2;
```

-- END Exercice 11_b

nombre_films_store_1	nombre_films_store_2
759	762

-- Exercice 12_a

drop function if exists func_ex12;

delimiter //

create function func_ex12 (id_store int)

returns int

begin

declare nombre int;

select

count(*)

into nombre

from

customer

where

store_id = id_store;

return nombre;

end; //

delimiter ;

-- END Exercice 12_a**-- Exercice 12_b**

select

```
func_ex12(1) as nombre_clients_store_1,
func_ex12(2) as nombre_clients_store_2;
```

-- END Exercice 12_b

nombre_clients_store_1	nombre_clients_store_2
325	273

-- Exercice 13_a

drop function if exists func_ex13;

delimiter //

create function func_ex13 (id_store int)

returns double

begin

declare montant double;

select

sum(amount)

into montant

from

payment

inner join

staff

on

payment.staff_id = staff.staff_id

where

staff.store_id = id_store;

```
return montant;
end; //
```

```
delimiter ;
```

```
-- END Exercise 13_a
```

```
-- Exercise 13_b
```

```
select
```

```
    func_ex13(1) as revenu_total_store_1,
```

```
    func_ex13(2) as revenu_total_store_2;
```

```
-- END Exercise 13_b
```

revenu_total_store_1	revenu_total_store_2
33490.55	33927.04

```
-- Exercise 14_a
```

```
drop procedure if exists proc_ex14;
```

```
delimiter //
```

```
create procedure proc_ex14 ()
```

```
begin
```

```
update film
```

```
set last_update = NOW();
```

```
end; //
```

```
delimiter ;
```

```
-- END Exercise 14_a
```

```
-- Exercise 14_b
```

```
call proc_ex14();
```

```
select
```

```
    film_id,
```

```
    title,
```

```
    last_update
```

```
from film
```

```
limit 20;
```

```
-- END Exercise 14_b
```

film_id	title	last_update	film_id	title	last_update
1	ACADEMY DINOSAUR	2006-02-15 05:03:42	1	ACADEMY DINOSAUR	2015-12-18 17:19:52
2	ACE GOLDFINGER	2006-02-15 05:03:42	2	ACE GOLDFINGER	2015-12-18 17:19:52
3	ADAPTATION HOLES	2006-02-15 05:03:42	3	ADAPTATION HOLES	2015-12-18 17:19:52
4	AFFAIR PREJUDICE	2006-02-15 05:03:42	4	AFFAIR PREJUDICE	2015-12-18 17:19:52
5	AFRICAN EGG	2006-02-15 05:03:42	5	AFRICAN EGG	2015-12-18 17:19:52
6	AGENT TRUMAN	2006-02-15 05:03:42	6	AGENT TRUMAN	2015-12-18 17:19:52
7	AIRPLANE SIERRA	2006-02-15 05:03:42	7	AIRPLANE SIERRA	2015-12-18 17:19:52
8	AIRPORT POLLOCK	2006-02-15 05:03:42	8	AIRPORT POLLOCK	2015-12-18 17:19:52
9	ALABAMA DEVIL	2006-02-15 05:03:42	9	ALABAMA DEVIL	2015-12-18 17:19:52
10	ALADDIN CALENDAR	2006-02-15 05:03:42	10	ALADDIN CALENDAR	2015-12-18 17:19:52
11	ALAMO VIDEOTAPE	2006-02-15 05:03:42	11	ALAMO VIDEOTAPE	2015-12-18 17:19:52
12	ALASKA PHANTOM	2006-02-15 05:03:42	12	ALASKA PHANTOM	2015-12-18 17:19:52
13	ALI FOREVER	2006-02-15 05:03:42	13	ALI FOREVER	2015-12-18 17:19:52
14	ALICE FANTASIA	2006-02-15 05:03:42	14	ALICE FANTASIA	2015-12-18 17:19:52
15	ALIEN CENTER	2006-02-15 05:03:42	15	ALIEN CENTER	2015-12-18 17:19:52
16	ALLEY EVOLUTION	2006-02-15 05:03:42	16	ALLEY EVOLUTION	2015-12-18 17:19:52
17	ALONE TRIP	2006-02-15 05:03:42	17	ALONE TRIP	2015-12-18 17:19:52
18	ALTER VICTORY	2006-02-15 05:03:42	18	ALTER VICTORY	2015-12-18 17:19:52
19	AMADEUS HOLY	2006-02-15 05:03:42	19	AMADEUS HOLY	2015-12-18 17:19:52
20	AMELIE HELLFIGHTERS	2006-02-15 05:03:42	20	AMELIE HELLFIGHTERS	2015-12-18 17:19:52

-- Exercice 15_a

```
drop procedure if exists proc_ex15;
delimiter //
create procedure proc_ex15 (
    in id_store int,
    out nombre_locations int,
    out nombre_exemplaires int)
begin
select
    count(rental_id)
into nombre_locations
from
    rental
inner join
    inventory
on
    rental.inventory_id = inventory.inventory_id
where
    store_id = id_store;

select
    count(inventory_id)
into nombre_exemplaires
from
    inventory
where
    store_id = id_store;

end; //
```

delimiter ;

-- END Exercice 15_a**-- Exercice 15_b**

```
call proc_ex15(1,@nombre_locations, @nombre_exemplaires);
select
```

```
    @nombre_locations,
    @nombre_exemplaires;
```

-- END Exercice 15_b

@nombre_locations	@nombre_exemplaires
7933	2270