1. Pasar la base cajainde\_bd a una de prueba en el servidor de sindicato con la tarea asignada
2. Aplicar los siguientes updates dentro de la base de prueba el cual asigna el valor de la columna solicitudes.sol\_numero en la columna solicitud\_avales.sol\_ava\_solicitud

update solicitud\_avales

inner join solicitudes on solicitudes.sol\_id = solicitud\_avales.sol\_ava\_solicitud

set sol\_ava\_solicitud = solicitudes.sol\_numero

1. Sobre la base de prueba aplicar el siguiente update, el cual asigna el valor de la columna solicitudes.sol\_numero en la columna imagenes.ima\_solicitud

update imagenes inner join solicitudes on sol\_id = ima\_solicitud

set ima\_solicitud=sol\_numero;

1. Sobre la base de prueba aplicar el siguiente update, el cual asigna el valor de la columna solicitudes.sol\_numero en la columna creditos\_final.cre\_solicitud

update creditos\_final inner join solicitudes on sol\_id = cre\_solicitud set cre\_solicitud = sol\_numero;

1. Sobre la base de prueba aplicar el siguiente update, el cual asigna el valor de la columna solicitudes.sol\_numero en la columna solicitudes.sol\_id

update solicitudes set sol\_id = sol\_numero;

1. El campo estatus db en todas las solicitudes anteriores que ya estén aceptadas o tengan un estatus más alto a pendiente debe ser 2

update solicitudes set sol\_estatus\_db=**2** where sol\_estatus > **1**;

update solicitudes set sol\_estatus\_db=**1** where sol\_estatus =**1**;

1. Verificar que no haya avales en la tabla solicitud\_avales que tengan el sol\_ava\_solicitud nulo, si estos tienen créditos que se actualice la solicitud, si no tiene sol ni crédito, que los elimine

delete solicitud\_avales from solicitud\_avales LEFT JOIN solicitudes ON sol\_ava\_solicitud=sol\_id

WHERE sol\_id IS NULL;

1. Verificar que no haya imágenes de la tabla imagnes que tengan el ima\_solicitud nulo, si estas tienen crédito que se actualice la solicitud, si no tiene sol ni crédito, que los elimine

delete imagenes from imagenes left join solicitudes on imagenes.ima\_solicitud = sol\_id where sol\_id is null

1. ima\_estatus siempre debe tener un valor en todas las solicitudes

select \* from imagenes where ima\_estatus is null

1. Que todas las amortizaciones tengan amo\_usu\_id, si no, aplicar la sig consulta

update amortizacion inner join creditos\_final on cre\_id = amo\_credito

set amo\_usu\_id = cre\_usu\_id where amo\_usu\_id is null

1. Truncar toda la base de Sindicato
2. Deshabilitar los constraints con la siguiente sentencia

SET GLOBAL FOREIGN\_KEY\_CHECKS=**0**;

1. Pasar los siguientes catalogos del ambiente local de mi maquina
   1. AMORTIZACION\_ESTATUS
   2. BANCOS\_CONCEPTOS
   3. CREDITO\_ESTATUS
   4. PRODUCTOS
   5. PAGOS\_ESTATUS
   6. ROLES
   7. SOLICITUD\_ESTATUS
2. Pasar del ambiente de cajainde\_bd primero las siguientes tablas

EMPRESAS

IMÁGENES

SOLICITUDES

USUARIOS

SOLICITUD\_ AVALES

1. Pasar las tablas restantes de cajainde\_bd de producción al ambiente de sindicato, EXCEPTUANDO las siguientes tablas, pasar primero las tablas de solicitudes y todas sus dependencias
   1. AMORTIZACION\_ESTATUS
   2. BANCOS\_CONCEPTOS
   3. BANCO\_AJUSTES
   4. BANCO\_EDOCTA
   5. BANCOS
   6. ESTADO\_CUENTA
   7. PRODUCTOS
   8. PAGOS\_ESTATUS
   9. ROLES
   10. SOLICITUD\_ESTATUS
2. Uniformar los estatus de amortización, hay que verificar que ya no haya amortizaciones sin amo\_estatus\_int y que todos los estatus en letra estén uniformados

Select distinct(amo\_estatus) from amortizacion;

update amortizacion set amo\_estatus ="DEUDA\_FINIQUITO" where amo\_estatus like "%FINIQ%";

update amortizacion set amo\_estatus = "PAGADO" where amo\_estatus like "%PAGADO%";

1. Actualizar el amo\_estatus\_int conforme el amo\_estatus

update amortizacion inner join amortizacion\_estatus

on amortizacion\_estatus.amo\_est\_nombre = amortizacion.amo\_estatus

set amo\_estatus\_int = amortizacion\_estatus.amo\_est\_id

where amo\_estatus\_int is null

1. A todos los estatus de amortizacion que tienen créditos con estatus cerrados (Cancelados, Incobrables)

#Actualiza los estatus de amortizacion a Cancelados cuando el credito esta cancelado

#y la amortizacion pendiente

update amortizacion inner join creditos\_final on cre\_id = amo\_credito

set amo\_estatus\_int = **14** where cre\_estatus = **3** and amo\_estatus\_int = **1**;

#Actualiza los estatus de amortizacion a incobrables cuando el credito esta incobrable

#y la amortizacion pendiente

update amortizacion inner join creditos\_final on cre\_id = amo\_credito

set amo\_estatus\_int = **12** where cre\_estatus = **5** and amo\_estatus\_int = **1**;

1. Actualizar el usu\_estatus con el siguiente update

update usuarios set usu\_estatus = **0** where usu\_fecha\_baja is not null;

update usuarios set usu\_estatus = **1** where usu\_fecha\_baja is null;

1. Se modifican todos los estatus de solicitudes para adaptarlos a los nuevos estatus:

##Asigna el estatus 8 doc aprobada a las solicitudes que tienen los documentos

##entregados pero aun no tienen hecho el deposito

update solicitudes, creditos\_final set sol\_estatus = **8**

where cre\_solicitud = sol\_id and cre\_fecha\_deposito is null and sol\_estatus = **5**;

##Asigna el estatus Cerrada, a las solicitudes que tienen el deposito hecho y los documentos entregados

update solicitudes inner join creditos\_final on cre\_solicitud = sol\_id

set sol\_estatus = **6**

where cre\_fecha\_deposito is not null and sol\_estatus = **5**;

##Estatus rechazada

update solicitudes set sol\_estatus = **7** where sol\_estatus in (**4**);

##Estatus fondeado

Update solicitudes set sol\_estatus = **4** where sol\_estatus in (**3**);

##Estatus autorizado

Update solicitudes set sol\_estatus = **3** where sol\_estatus in (**2**);

##Se ponen en estatus autorizados las imagenes y los avales de la solicitud

update solicitudes inner join solicitud\_avales on sol\_ava\_solicitud = sol\_id

inner join imagenes on ima\_solicitud = sol\_id

set sol\_ava\_estatus = **3**, ima\_estatus= **3** where sol\_estatus in (**3**);

##TODAS LAS SOLICITUDES ANTERIORES QUE TENGAN ESTATUS 1 SE PONDRÁN CON EL NUEVO ESTATUS 2 VALIDANDO

update solicitudes set sol\_estatus = **2** where sol\_estatus in (**1**);

##Aqui nada mas poner los estatus correctos

update solicitudes inner join solicitud\_avales on sol\_ava\_solicitud = sol\_id

inner join imagenes on ima\_solicitud = sol\_id

set sol\_ava\_estatus = **2**, ima\_estatus= **2** where sol\_estatus in (**2**);

1. Se tendrán que asignar los estatus de las bajas de los usuarios que ya se han dado de baja, para determinar quienes tienen ahorro pendiente por devolver y quienes tienen créditos por devolver

##ASIGNA ESTATUS 1 DE BAJA

update baja\_empleados , (

select bae\_id, usu\_id, IFNULL(creditos.saldoPendiente,0) as deudaCreditos, IFNULL(movimientos.totalMovimientos,0) as totalMovimientos, bae\_estatus

from baja\_empleados inner join usuarios on usu\_id = bae\_id\_empleado

left join

(

select mov\_usu\_id, sum(mov\_deposito) as totalMovimientos from movimientos group by mov\_usu\_id

) movimientos on usu\_id = movimientos.mov\_usu\_id

left join (

select amo\_usu\_id, sum(amo\_monto\_pago) as saldoPendiente from amortizacion

where amo\_estatus\_int = 1

group by amo\_usu\_id

) creditos on creditos.amo\_usu\_id = usu\_id

) bae2 set baja\_empleados.bae\_ahorros = bae2.totalMovimientos, baja\_empleados.bae\_deuda\_creditos=bae2.deudaCreditos, baja\_empleados.bae\_estatus=1

where bae2.bae\_id = baja\_empleados.bae\_id

and (bae2.totalMovimientos >= 1 and bae2.deudaCreditos >= 1

or bae2.totalMovimientos <1 and bae2.deudaCreditos > 1);

##ASIGNA ESTATUS 2 DE BAJA

update baja\_empleados , (

select bae\_id, usu\_id, IFNULL(creditos.saldoPendiente,0) as deudaCreditos, IFNULL(movimientos.totalMovimientos,0) as totalMovimientos, bae\_estatus

from baja\_empleados inner join usuarios on usu\_id = bae\_id\_empleado

left join

(

select mov\_usu\_id, sum(mov\_deposito) as totalMovimientos from movimientos group by mov\_usu\_id

) movimientos on usu\_id = movimientos.mov\_usu\_id

left join (

select amo\_usu\_id, sum(amo\_monto\_pago) as saldoPendiente from amortizacion

where amo\_estatus\_int = 1

group by amo\_usu\_id

) creditos on creditos.amo\_usu\_id = usu\_id

) bae2 set baja\_empleados.bae\_ahorros = bae2.totalMovimientos, baja\_empleados.bae\_deuda\_creditos=bae2.deudaCreditos, baja\_empleados.bae\_estatus=2

where bae2.bae\_id = baja\_empleados.bae\_id

and bae2.totalMovimientos >= 1 and bae2.deudaCreditos < 1;

##ASIGNA ESTATUS 3 DE BAJA

update baja\_empleados , (

select bae\_id, usu\_id, IFNULL(creditos.saldoPendiente,0) as deudaCreditos, IFNULL(movimientos.totalMovimientos,0) as totalMovimientos, bae\_estatus

from baja\_empleados inner join usuarios on usu\_id = bae\_id\_empleado

left join

(

select mov\_usu\_id, sum(mov\_deposito) as totalMovimientos from movimientos group by mov\_usu\_id

) movimientos on usu\_id = movimientos.mov\_usu\_id

left join (

select amo\_usu\_id, sum(amo\_monto\_pago) as saldoPendiente from amortizacion

where amo\_estatus\_int = 1

group by amo\_usu\_id

) creditos on creditos.amo\_usu\_id = usu\_id

) bae2 set baja\_empleados.bae\_ahorros = bae2.totalMovimientos, baja\_empleados.bae\_deuda\_creditos=bae2.deudaCreditos, baja\_empleados.bae\_estatus=3

where bae2.bae\_id = baja\_empleados.bae\_id

and bae2.totalMovimientos < 10 and bae2.deudaCreditos < 10;

1. Se introducen en la tabla de bancos todos los conceptos anteriores a la fecha de entrega

##Archivo de pagos

INSERT INTO bancos (ban\_concepto, ban\_monto, ban\_empresa, ban\_fechatransaccion, ban\_id\_concepto\_sistema, ban\_ajustado)

SELECT 1, sum(pag\_deposito), pag\_empresa, pag\_fecha, pag\_arh\_id, 0

FROM pagos where pag\_arh\_id is not null

group by pag\_arh\_id;

##Archivo de aportaciones

INSERT INTO bancos (ban\_concepto, ban\_monto, ban\_empresa, ban\_fechatransaccion, ban\_id\_concepto\_sistema, ban\_ajustado)

SELECT 2, sum(mov\_deposito), mov\_empresa, mov\_fecha, mov\_arh\_id, 0

FROM movimientos where mov\_arh\_id is not null

group by mov\_arh\_id;

##PAGO EXTEMPORANEO

INSERT INTO bancos (ban\_concepto, ban\_monto, ban\_empresa, ban\_fechatransaccion, ban\_id\_concepto\_sistema, ban\_ajustado)

SELECT 4, pag\_deposito, pag\_empresa, pag\_fecha, pag\_id, 0

FROM pagos where pag\_estatus = 10;

## PAGO A CAPITAL

INSERT INTO bancos (ban\_concepto, ban\_monto, ban\_empresa, ban\_fechatransaccion, ban\_id\_concepto\_sistema, ban\_ajustado)

SELECT 5, pag\_deposito, pag\_empresa, pag\_fecha, pag\_id, 0

FROM pagos where pag\_estatus = 9;

## DEVOLUCION ACUMULADO

INSERT INTO bancos (ban\_concepto, ban\_monto, ban\_empresa, ban\_fechatransaccion, ban\_id\_concepto\_sistema, ban\_ajustado)

SELECT 6, pag\_deposito, pag\_empresa, pag\_fecha, pag\_id, 0

FROM pagos where pag\_estatus = 11;

## APORTACION VOLUNTARIA

INSERT INTO bancos (ban\_concepto, ban\_monto, ban\_empresa, ban\_fechatransaccion, ban\_id\_concepto\_sistema, ban\_ajustado)

SELECT 7, mov\_deposito, mov\_empresa, mov\_fecha, mov\_id, 0

FROM movimientos where mov\_producto = 3 and mov\_ar = 1;

## DEVOLUCION AHORRO FIJO

INSERT INTO bancos (ban\_concepto, ban\_monto, ban\_empresa, ban\_fechatransaccion, ban\_id\_concepto\_sistema, ban\_ajustado)

SELECT 8, mov\_deposito, mov\_empresa, mov\_fecha, mov\_id, 0

FROM movimientos where mov\_producto = 1 and mov\_ar = 1 and mov\_deposito<0;

## DEVOLUCION AHORRO NO FIJO

INSERT INTO bancos (ban\_concepto, ban\_monto, ban\_empresa, ban\_fechatransaccion, ban\_id\_concepto\_sistema, ban\_ajustado)

SELECT 9, mov\_deposito, mov\_empresa, mov\_fecha, mov\_id, 0

FROM movimientos where mov\_producto = 2 and mov\_ar = 1 and mov\_deposito<0;

## DEVOLUCION AHORRO VOLUNTARIO

INSERT INTO bancos (ban\_concepto, ban\_monto, ban\_empresa, ban\_fechatransaccion, ban\_id\_concepto\_sistema, ban\_ajustado)

SELECT 10, mov\_deposito, mov\_empresa, mov\_fecha, mov\_id, 0

FROM movimientos where mov\_producto = 3 and mov\_ar = 1 and mov\_deposito<0;

## DEVOLUCION RENDIMIENTO AHORRO FIJO

INSERT INTO bancos (ban\_concepto, ban\_monto, ban\_empresa, ban\_fechatransaccion, ban\_id\_concepto\_sistema, ban\_ajustado)

SELECT 11, mov\_deposito, mov\_empresa, mov\_fecha, mov\_id, 0

FROM movimientos where mov\_producto = 1 and mov\_ar = 2 and mov\_deposito<0;

## DEPOSITO CREDITOS

INSERT INTO bancos (ban\_concepto, ban\_monto, ban\_empresa, ban\_fechatransaccion, ban\_id\_concepto\_sistema, ban\_ajustado)

SELECT 12, cre\_prestamo, usu\_empresa, cre\_fecha\_deposito, cre\_id, 0

FROM creditos\_final inner join usuarios on usu\_id = cre\_usu\_id

where cre\_prestamo > 0 and cre\_prestamo is not null;

##DESCUENTO X COBRAR DE CREDITO

INSERT INTO bancos (ban\_concepto, ban\_monto, ban\_empresa, ban\_fechatransaccion, ban\_id\_concepto\_sistema, ban\_ajustado)

SELECT 13, pag\_deposito, pag\_empresa, pag\_fecha, pag\_id, 0

FROM pagos where pag\_estatus = 12;

## DEVOLUCION RENDIMIENTO AHORRO NO FIJO

INSERT INTO bancos (ban\_concepto, ban\_monto, ban\_empresa, ban\_fechatransaccion, ban\_id\_concepto\_sistema, ban\_ajustado)

SELECT 14, mov\_deposito, mov\_empresa, mov\_fecha, mov\_id, 0

FROM movimientos where mov\_producto = 2 and mov\_ar = 2 and mov\_deposito<0;

## DEVOLUCION RENDIMIENTO AHORRO VOLUNTARIO

INSERT INTO bancos (ban\_concepto, ban\_monto, ban\_empresa, ban\_fechatransaccion, ban\_id\_concepto\_sistema, ban\_ajustado)

SELECT 15, mov\_deposito, mov\_empresa, mov\_fecha, mov\_id, 0

FROM movimientos where mov\_producto = 3 and mov\_ar = 2 and mov\_deposito<0;

1. Habilitar los constraints de toda la base

SET GLOBAL FOREIGN\_KEY\_CHECKS=**1**;

1. Asignarle el usu\_rol a todos los usuarios

update usuarios set usu\_rol = **3** where usu\_rol is null

1. Asignar el valor de autoincrement correcto

##Consultas para conocer detalles de una tabla, entre ellos el autoincrement

select AUTO\_INCREMENT from information\_schema.TABLES where TABLE\_SCHEMA='sindicato' and TABLE\_NAME='solicitudes';

SHOW TABLE STATUS LIKE 'solicitudes'

## ACCIONES A REALIZAR DESPUES DE LA MIGRACION

Verificar la lista de empleados pendientes por dar de alta

Verificar el reporte de morosos que coincida con el de caja\_indpendencia

Verificar solicitudes\_pendientes

Verificar a los usuarios dados de baja, que esten en el estatus correcto

Hacer conteos de cuantos registros tienen cre\_solicitud en ambas bases

$2,024

$ 1,449.91

3706