Application Development II

Time Designing, Writing, and Debugging

10 hours (both Melchisedek and Colten)

Colten Shipe and Melchisedek Dulcio

CpS 301

November 16, 2020

# Program Report

## Academic Integrity Report

By affixing my signature below, I certify that the accompanying work represents my own intellectual effort. Furthermore, I have received no outside help other than what is documented below and/or in program source code comments.

\_\_\_\_Colten Shipe\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Signature

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Name / Email / URL** | **Nature of help** | **Time spent** |
|  |  |  |  |
|  |  |  |  |

# Procedure

CREATE DEFINER=`root`@`localhost` PROCEDURE `create\_service`(IN service\_datetime DATETIME, IN theme VARCHAR(40), in person\_id int, in tmpltsvc\_id int, out error\_msg int, out newsvc\_id int)

begin

declare chk\_id int;

declare svc\_id int;

declare newitem\_id int;

SELECT

Service\_ID

INTO chk\_id FROM

service

WHERE

Svc\_DateTime = service\_datetime;

if chk\_id is not null then

set error\_msg = 1;

else

select max(Service\_ID) + 1 into svc\_id from service;

INSERT INTO `wsoapp2`.`service` (`Service\_ID`, `Svc\_DateTime`,`Theme\_Event`)

VALUES (svc\_id, service\_datetime, theme);

if person\_id is not null then

insert into `wsoapp2`.`fills\_role`(`Person\_ID`,`Service\_ID`,`Role\_Type`,`Confirmed`)

values(person\_id, svc\_id, 'S', 'Y');

end if;

if tmpltsvc\_id is not null then

select max(Service\_Item\_ID) into newitem\_id from service\_item;

insert into service\_item

select @newitem\_id + 1, svc\_id, Seq\_Num, Event\_Type\_ID, null, null, 'Y', null, null, null

from service\_item where Service\_ID = tmpltsvc\_id;

end if;

set newsvc\_id = svc\_id;

set error\_msg = 0;

end if;

end

# Views

## SongUsageView

CREATE

ALGORITHM = UNDEFINED

DEFINER = `root`@`localhost`

SQL SECURITY DEFINER

VIEW `songusageview` AS

SELECT

`song`.`Song\_ID` AS `Song\_ID`,

`song`.`Song\_Type` AS `Song\_Type`,

`song`.`Title` AS `Title`,

`song`.`Hymnbook\_Num` AS `Hymnbook\_Num`,

`song`.`Arranger` AS `Arranger`,

MAX(`service`.`Svc\_DateTime`) AS `LastUsedDate`

FROM

((`song`

LEFT JOIN `service\_item` ON ((`song`.`Song\_ID` = `service\_item`.`Song\_ID`)))

LEFT JOIN `service` ON ((`service`.`Service\_ID` = `service\_item`.`Service\_ID`)))

WHERE

(`song`.`Song\_Type` <> 'C')

GROUP BY `song`.`Song\_ID` , `song`.`Song\_Type` , `song`.`Title` , `song`.`Hymnbook\_Num` , `song`.`Arranger`

ORDER BY `LastUsedDate` , `song`.`Title`

## service\_view

CREATE

ALGORITHM = UNDEFINED

DEFINER = `root`@`localhost`

SQL SECURITY DEFINER

VIEW `service\_view` AS

SELECT

`service`.`Service\_ID` AS `Service\_ID`,

`service`.`Svc\_DateTime` AS `Svc\_DateTime`,

`service`.`Theme\_Event` AS `theme`,

`song\_leader\_view`.`songleader` AS `songleader`,

`organist\_view`.`organist` AS `organist`,

`pianist\_view`.`pianist` AS `pianist`,

`service\_item`.`Seq\_Num` AS `Seq\_Num`,

`event\_type`.`Description` AS `event`,

(CASE

WHEN

((`song`.`Title` IS NOT NULL)

AND (`song`.`Song\_Type` = 'H'))

THEN

CONCAT(COALESCE(`song`.`Hymnbook\_Num`, ''),

' - ',

`song`.`Title`)

WHEN

((`song`.`Title` IS NOT NULL)

AND (`song`.`Song\_Type` = 'C'))

THEN

`song`.`Title`

ELSE `service\_item`.`Title`

END) AS `title`,

(CASE

WHEN `service\_item`.`Ensemble\_ID` THEN `ensemble`.`Name`

WHEN

`service\_item`.`Person\_ID`

THEN

CONCAT(`person`.`First\_Name`,

' ',

`person`.`Last\_Name`)

END) AS `name`,

`service\_item`.`Notes` AS `Notes`

FROM

((((((((`service\_item`

JOIN `service` ON ((`service\_item`.`Service\_ID` = `service`.`Service\_ID`)))

LEFT JOIN `event\_type` ON ((`service\_item`.`Event\_Type\_ID` = `event\_type`.`Event\_Type\_ID`)))

LEFT JOIN `person` ON ((`service\_item`.`Person\_ID` = `person`.`Person\_ID`)))

LEFT JOIN `ensemble` ON ((`service\_item`.`Ensemble\_ID` = `ensemble`.`Ensemble\_ID`)))

LEFT JOIN `song` ON ((`song`.`Song\_ID` = `service\_item`.`Song\_ID`)))

LEFT JOIN `song\_leader\_view` ON ((`song\_leader\_view`.`Service\_ID` = `service\_item`.`Service\_ID`)))

LEFT JOIN `organist\_view` ON ((`organist\_view`.`Service\_ID` = `service\_item`.`Service\_ID`)))

LEFT JOIN `pianist\_view` ON ((`pianist\_view`.`Service\_ID` = `service\_item`.`Service\_ID`)))