



Group 2 Presentation

Summer Camp Operations Management System

Database Design & Implementation

Robert Marshall
Ram Ganesan
Anuj Joshi
Pierre Augustamar
December 15, 2017

Topics

- Database Purpose
- Business Problems Addressed
- Who is our audience
- Application Usage
- Design Logic or workflow
- Design Diagram
- User Roles
- System Outlook
- System Performance
- System Logging/Monitoring
- Demonstration/walk-through
- Additional Enhancements
- Project Dynamics
- Lessons Learned

Database Purpose

- System to maintain students' records, courses, and activities for which they are registered for the Summer.
- System to provide information regarding the teachers that are assigned to specific classes and the locations that these courses will be taken place.
- Administrators and staff should be able to retrieve information for a particular course or a specific student.
- Only faculty and administrators are allowed to make changes to the data.
- Students and parents will be given access to view data that are specific to student courses and activities.

Business Problems Addressed

- Display weekly scheduled activities
- Display staff and student's location
- Display student's contact information
- Notify administrators when verification forms are out of date
- Notify administrators when verification forms are unassigned or malformed
- Notify administrators when payments were not received

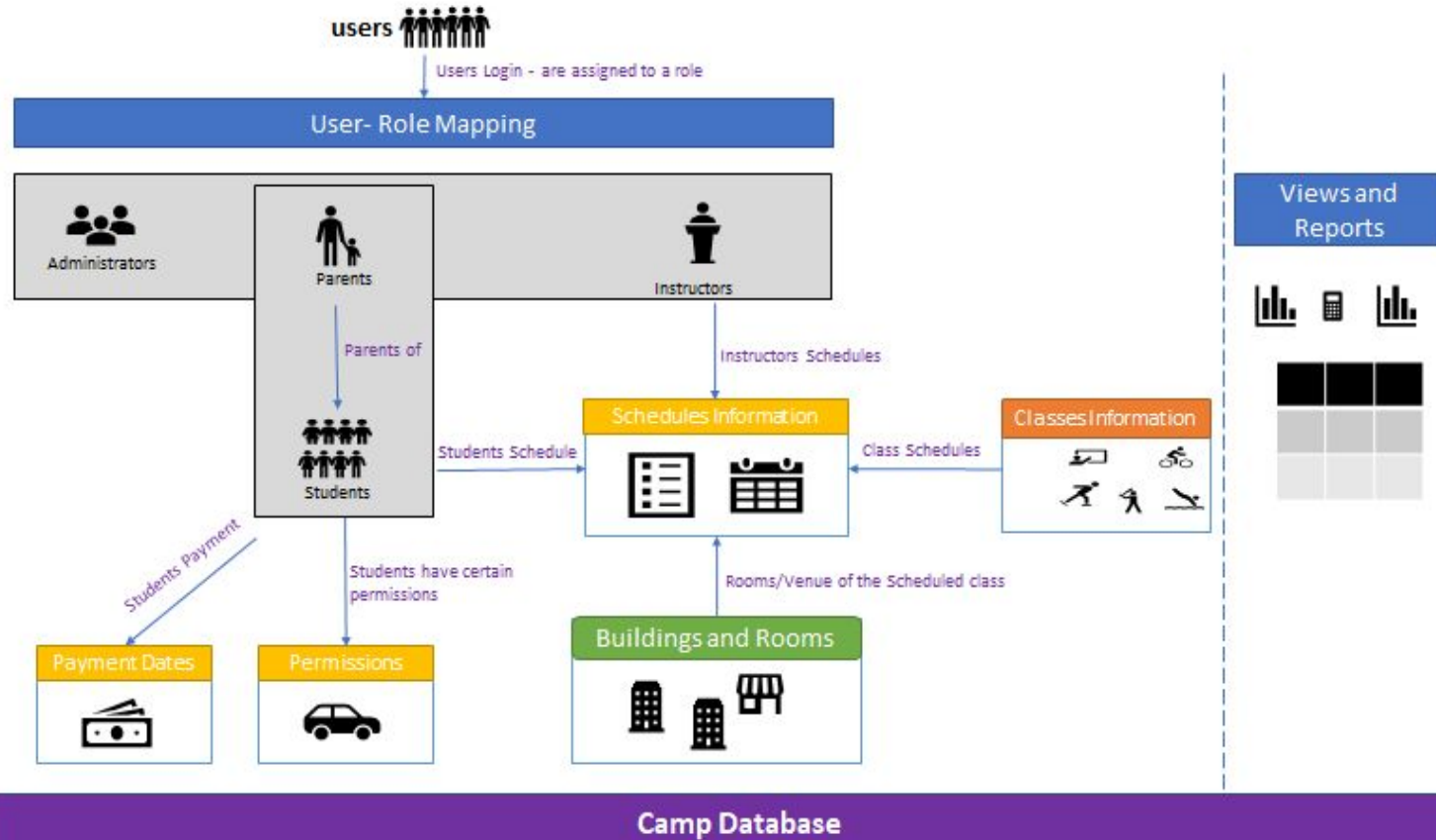
Who is our audience?

- Targets two main groups:
 - Children between the ages of 10-18
 - Working parents who have schooled aged kids
- Websites and online communities advertising for summer camps
- YMCA and other social community services needing to report on summer camp activities
- Low-income families needing a place for their kids during summer
- Local schools working with parents to find an alternative places for students during summer
- Local churches working with parishioners to find a place for kids to spend time during Summer
- Social services needing a summer camp for kids who are in transition from moving to different families

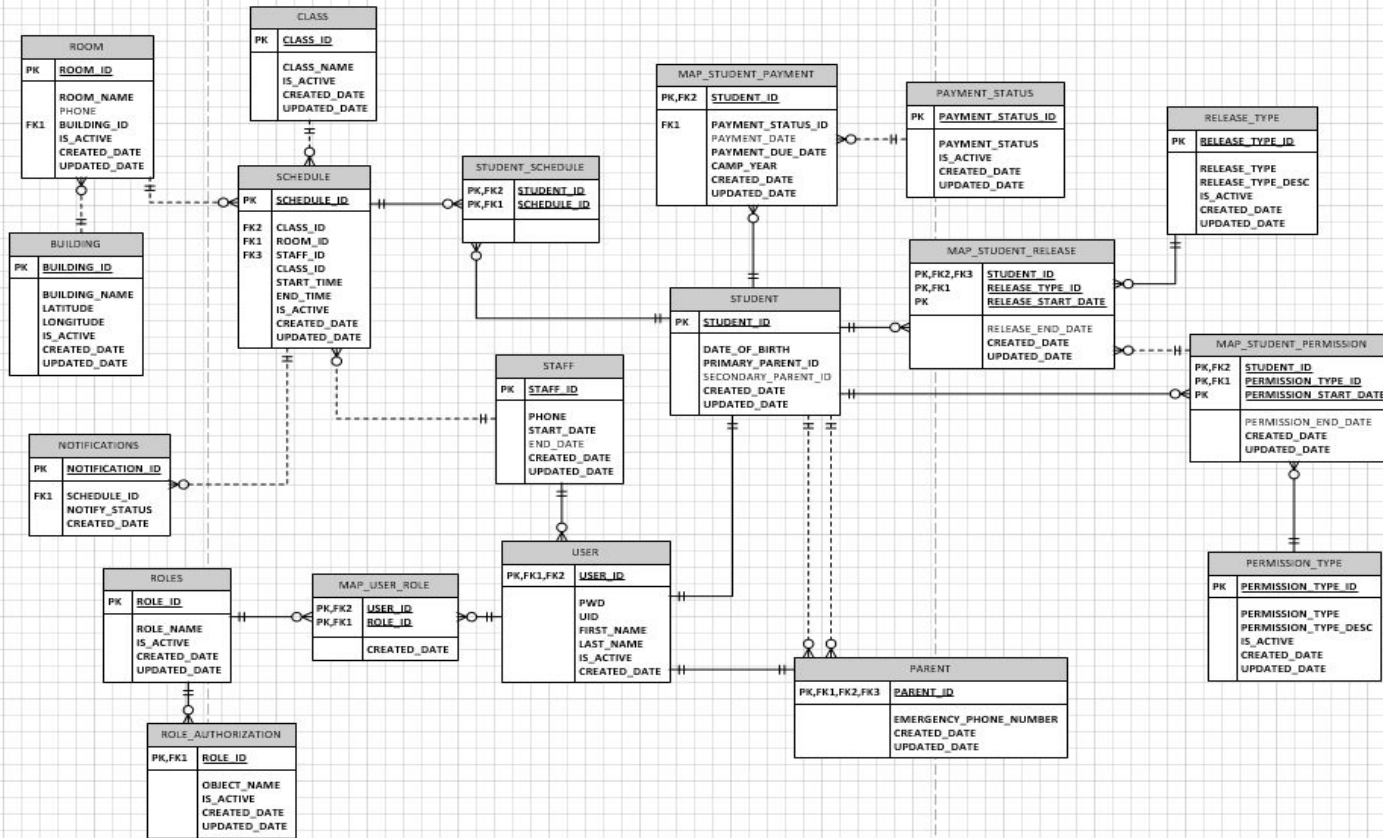
Application Usage

- Websites to manage a summer camp can use our database as the backend store
- Mobile apps can use our database as a backend store that will allow
 - parents to get summer camp notifications
 - kids to keep track of camp activities
- Advertising companies can use the features and business purposes to advertise for a specific summer camp
- Announcements and newsletters can be distributed to the list of users that are in the system

Workflow



Design Logic – Database Diagram



User Roles

- One of the requirements of this project was to define user/roles
- For this project we used the following roles:
 - Role 1 - administrator
 - Role 2 - instructor
 - Role 3 - work crews
 - Role 4 - parent
 - Role 5 - student
- Roles 1 - 3 are part of the staff
- Anyone except students can have multiple roles
- Someone with multiple roles will adopt the highest role.
- Users roles can be found through a built-in stored procedure
- Only active users will be eligible to use the system

User roles - continue

- Someone with role 1 can perform any activities in the system
- Someone with role 2 can perform his/her role including all the operations for roles 3 to 5
- Roles are tied with authorization.
- Views that are eligible to be accessed by parents are listed in a mapping table.
- Anyone with role 3 or work crews do not have access to any of the views or stored procedures in the system.

System Outlook

- System contains the following objects:
 - 19 user defined tables
 - 9 views
 - 3 functions
 - 3 triggers
 - 7 store procedures
 - 39 Primary keys and Foreign keys
- Schema name
 - All artifacts are created under the camp schema

System Outlook - continue

The screenshot below display detailed information about the tables created for the camp database

	name	object_id	principal_id	schema_id	parent_object_id	type	type_desc	create_date	modify_date	is_ms_shipped	is_published	is_schema_published
1	ROLES	245575913	NULL	5	0	U	USER_TABLE	2017-12-14 14:58:14.277	2017-12-14 14:58:14.560	0	0	0
2	USER	277576027	NULL	5	0	U	USER_TABLE	2017-12-14 14:58:14.283	2017-12-14 14:58:14.297	0	0	0
3	MAP_USER_ROLE	309576141	NULL	5	0	U	USER_TABLE	2017-12-14 14:58:14.283	2017-12-14 14:58:14.283	0	0	0
4	PARENT	357576312	NULL	5	0	U	USER_TABLE	2017-12-14 14:58:14.287	2017-12-14 14:58:14.290	0	0	0
5	STUDENT	405576483	NULL	5	0	U	USER_TABLE	2017-12-14 14:58:14.287	2017-12-14 14:58:14.307	0	0	0
6	STAFF	485576768	NULL	5	0	U	USER_TABLE	2017-12-14 14:58:14.290	2017-12-14 14:58:14.307	0	0	0
7	BUILDING	533576939	NULL	5	0	U	USER_TABLE	2017-12-14 14:58:14.290	2017-12-14 14:58:14.293	0	0	0
8	ROOM	565577053	NULL	5	0	U	USER_TABLE	2017-12-14 14:58:14.293	2017-12-14 14:58:14.307	0	0	0
9	CLASS	613577224	NULL	5	0	U	USER_TABLE	2017-12-14 14:58:14.293	2017-12-14 14:58:14.303	0	0	0
10	PAYMENT_STATUS	645577338	NULL	5	0	U	USER_TABLE	2017-12-14 14:58:14.293	2017-12-14 14:58:14.297	0	0	0
11	MAP_STUDENT_PAYMENT	677577452	NULL	5	0	U	USER_TABLE	2017-12-14 14:58:14.297	2017-12-14 14:58:14.297	0	0	0
12	PERMISSION_TYPE	741577680	NULL	5	0	U	USER_TABLE	2017-12-14 14:58:14.297	2017-12-14 14:58:14.300	0	0	0
13	RELEASE_TYPE	773577794	NULL	5	0	U	USER_TABLE	2017-12-14 14:58:14.300	2017-12-14 14:58:14.303	0	0	0
14	MAP_STUDENT_PERMISSION	805577908	NULL	5	0	U	USER_TABLE	2017-12-14 14:58:14.300	2017-12-14 14:58:14.300	0	0	0
15	MAP_STUDENT_RELEASE	869578136	NULL	5	0	U	USER_TABLE	2017-12-14 14:58:14.300	2017-12-14 14:58:14.300	0	0	0
16	SCHEDULE	933578364	NULL	5	0	U	USER_TABLE	2017-12-14 14:58:14.303	2017-12-14 14:58:14.360	0	0	0
17	STUDENT_SCHEDULE	1029578706	NULL	5	0	U	USER_TABLE	2017-12-14 14:58:14.307	2017-12-14 14:58:14.307	0	0	0
18	NOTIFICATIONS	1093578934	NULL	5	0	U	USER_TABLE	2017-12-14 14:58:14.360	2017-12-14 14:58:14.360	0	0	0
19	ROLE_AUTHORIZATION	1141579105	NULL	5	0	U	USER_TABLE	2017-12-14 14:58:14.427	2017-12-14 14:58:14.560	0	0	0

The screenshot below display detailed information about views, stored procedures, and functions created for the camp database

type	name
functions	fn_Check_Permission
functions	fn_GetCurrent_Class
functions	fn_StudentWithMultiPermissions
procedures	usp_GetCurrentBuildingActivity
procedures	uspGetReportingStudentAndAssignedClasses
procedures	uspGetReportingTeacherAndAssignedClasses
procedures	usp_getstudentpermissions
procedures	usp_getreleaseinformation
procedures	usp_getpaymentinformation
procedures	usp_GetChildDetails
triggers	TRG_NOTIFY_SCHEDULE_CHANGE
triggers	USER_DELETE
triggers	STUDENT_PAYMENT_UPDATE
views	vwUserRoles
views	vwStudentsSchedules
views	vwTeachersSchedules
views	vwCurrentActiveUsers
views	vwCurrentInactiveUsers
views	vw_getpaymentinformation
views	vw_getstudentpermissions
views	vw_getreleaseinformation
views	vw_getweeklyschedules

System Performance

- System is designed to support on average 120 students, 10 teachers, 5 administrators, and 20 work crews on a giving summer camp year
- System can handle multiple requests a second
- Very little write on the system during camp activities, thus, minimal possibilities of deadlock or blocking
- Heavy reads are expected when camps are in session
- Select statements for each of the reports contain the needed columns
- Primary key constraints and Foreign key constraint to match relationships between the tables
- Queries are written based on the column's data structure. For instance, columns that are stored as date time are not converted when executing queries
- Queries are driven based on the existing clustered indexes
- Utilized the query analyzer to optimize our report queries
- In case of a database growth, the performance will remain stable because of the well-established indexes
- Ensured that no distinct, no order by clauses were used unless the functionality warrants it

System Performance - Continue

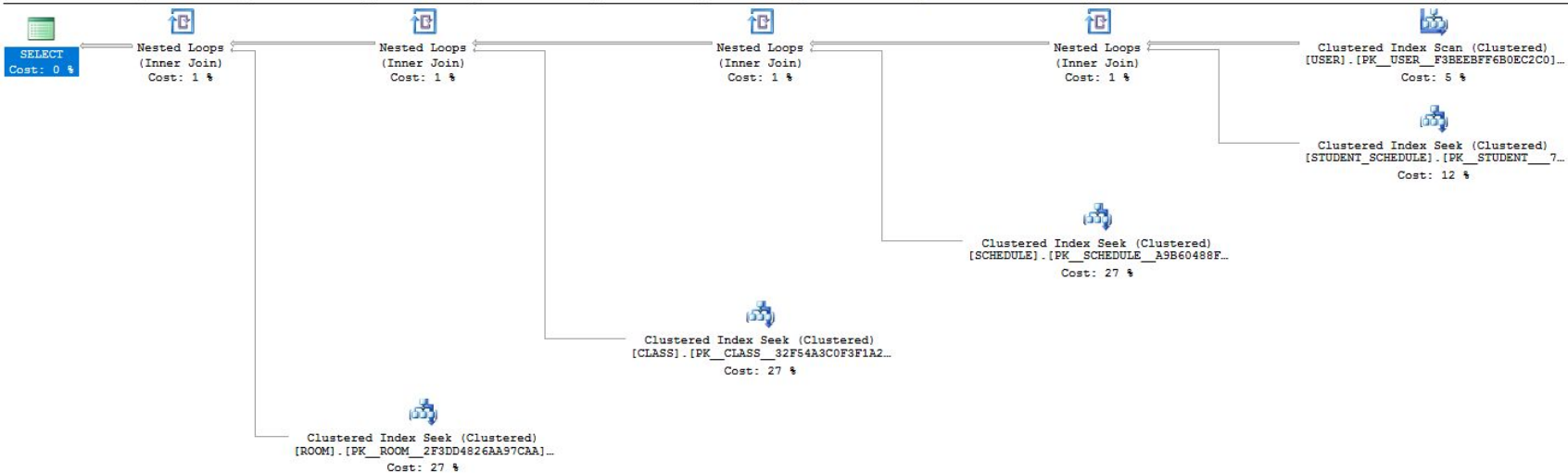
```
1
2
3 SELECT u.FIRST_NAME,u.LAST_NAME,c.CLASS_NAME,s.ROOM_ID,r.BUILDING_ID,s.START_TIME,s.END_TIME, s.CLASS_DATE
4 FROM [SUMMER_CAMP].[CAMP].[SCHEDULE] s
5 JOIN [SUMMER_CAMP].[CAMP].[STUDENT_SCHEDULE] su
6 ON su.SCHEDULE_ID = s.SCHEDULE_ID
7 JOIN [SUMMER_CAMP].[CAMP].[CLASS] c
8 ON c.CLASS_ID = s.CLASS_ID
9 JOIN [SUMMER_CAMP].[CAMP].[USER] u
10 ON u.USER_ID = su.STUDENT_ID
11 JOIN [SUMMER_CAMP].[CAMP].[ROOM] r
12 ON r.ROOM_ID = s.ROOM_ID
13 WHERE u.IS_ACTIVE = 1
```

100 %

Messages Execution plan

Query 1: Query cost (relative to the batch): 100%

SELECT u.FIRST_NAME,u.LAST_NAME,c.CLASS_NAME,s.ROOM_ID,r.BUILDING_ID,s.START_TIME,s.END_TIME, s.CLASS_DATE FROM [SUMMER_CAMP].[CAMP].[SCHEDULE] s JOIN [SUMMER_CAMP].[CAMP].[STUDENT_SCHEDULE] su...



System Performance - Continue - Explanation of the query

- Most queries only have roughly 100 records to process
- SQL Server performs "seeks" against the indexes, and it returns the data in about 30ms or less.
- No visible warnings are shown in the execution plan, thus no apparent issues with the joins for a complex query.

System Logging/Monitoring

- Tracking activities are done through analyzing last modified date
- Created date and updated date columns are added to the main tables
- Administrators can use the updated date to monitor changes in the system
- Updated date or last modified date are designed not to allow nulls so that changes can be tracked
- Updated date and created date can be leveraged by any downstream system for a delta load

Demonstration/walk-through

Use cases – Report showing a student's schedule

Artifact Name: CAMP.uspGetReportingStudentAndAssignedClasses

Object Type : Stored Procedure

Description : Reporting a specific student's schedule

Demo :

```
USE SUMMER_CAMP
DECLARE @UserId int
SELECT @UserId = 1
EXEC CAMP.uspGetReportingStudentAndAssignedClasses @UserId
```

	FIRST_NAME	LAST_NAME	CLASS_NAME	ROOM_ID	BUILDING_ID	START_TIME	END_TIME	CLASS_DATE
1	Sam	Mae	Animals	1	100	09:00:00.0000000	12:00:00.0000000	2018-06-11
2	Sam	Mae	Plants	2	100	09:00:00.0000000	12:00:00.0000000	2018-06-12
3	Sam	Mae	Survival	3	200	09:00:00.0000000	12:00:00.0000000	2018-06-13
4	Sam	Mae	Canoe	4	200	09:00:00.0000000	12:00:00.0000000	2018-06-14
5	Sam	Mae	Shelter	5	300	09:00:00.0000000	12:00:00.0000000	2018-06-15
6	Sam	Mae	Basket Weaving	10	500	13:00:00.0000000	16:00:00.0000000	2018-06-15
7	Sam	Mae	First Aid	9	500	13:00:00.0000000	16:00:00.0000000	2018-06-14
8	Sam	Mae	Reptiles	8	400	13:00:00.0000000	16:00:00.0000000	2018-06-13
9	Sam	Mae	Trees	7	400	13:00:00.0000000	16:00:00.0000000	2018-06-12
10	Sam	Mae	Insects	6	300	13:00:00.0000000	16:00:00.0000000	2018-06-11

Use cases – **Report showing a student's schedule - Continue**

- Most parents will execute this report to find out what classes/activities that their children will be having for each day during the camp session
- Administrators and instructors may also implement this report when trying to check on a specific student
- It is expected that this report will be executed at a lesser rate by the camp's staff

Use cases – Report showing an instructor's schedule

Artifact Name: CAMP.uspGetReportingTeacherAndAssignedClasses

Object Type : Stored Procedure

Description : Reporting the schedule activities/classes for a specific instructor

Demo :

```
USE SUMMER_CAMP  
DECLARE @StaffId int  
SELECT @StaffId = 512  
EXEC CAMP.uspGetReportingTeacherAndAssignedClasses @StaffId
```

FIRST_NAME	LAST_NAME	CLASS_NAME	ROOM_ID	BUILDING_ID	START_TIME	END_TIME	CLASS_DATE
Anuj	Joshi	Animals	1	100	09:00:00.0000000	12:00:00.0000000	2018-06-11
Anuj	Joshi	Animals	10	500	13:00:00.0000000	16:00:00.0000000	2018-06-11
Anuj	Joshi	Plants	2	100	09:00:00.0000000	12:00:00.0000000	2018-06-12
Anuj	Joshi	Plants	1	100	13:00:00.0000000	16:00:00.0000000	2018-06-12
Anuj	Joshi	Survival	3	200	09:00:00.0000000	12:00:00.0000000	2018-06-13
Anuj	Joshi	Basket Weaving	2	100	13:00:00.0000000	16:00:00.0000000	2018-06-13
Anuj	Joshi	Canoe	4	200	09:00:00.0000000	12:00:00.0000000	2018-06-14
Anuj	Joshi	Canoe	3	200	13:00:00.0000000	16:00:00.0000000	2018-06-14
Anuj	Joshi	Shelter	5	300	09:00:00.0000000	12:00:00.0000000	2018-06-15
Anuj	Joshi	Shelter	4	200	13:00:00.0000000	16:00:00.0000000	2018-06-15

Use cases – **Report showing an instructor's schedule**

- Report allows an instructor to check his/her schedule during the time this person has been teaching as part of a staff member at the camp
- Administrators can also execute this report to view the location and activities of a specific instructor

Use cases – Report showing a full list of every student's schedule

Artifact Name: CAMP.vwStudentsSchedules

Object Type : Views

Description : Reporting the schedule of all the students that are attending the Summer camp

Demo :

USE SUMMER_CAMP

SELECT * FROM CAMP.vwStudentsSchedules

FIRST_NAME	LAST_NAME	CLASS_NAME	ROOM_ID	BUILDING_ID	START_TIME	END_TIME	CLASS_DATE
Sam	Mae	Animals	1	100	09:00:00.0000000	12:00:00.0000000	2018-06-11
Sam	Mae	Plants	2	100	09:00:00.0000000	12:00:00.0000000	2018-06-12
Sam	Mae	Survival	3	200	09:00:00.0000000	12:00:00.0000000	2018-06-13
Sam	Mae	Canoe	4	200	09:00:00.0000000	12:00:00.0000000	2018-06-14
Sam	Mae	Shelter	5	300	09:00:00.0000000	12:00:00.0000000	2018-06-15
Jack	Paul	Insects	2	100	09:00:00.0000000	12:00:00.0000000	2018-06-11
Jack	Paul	Trees	3	200	09:00:00.0000000	12:00:00.0000000	2018-06-12
Jack	Paul	Animals	4	200	09:00:00.0000000	12:00:00.0000000	2018-06-13
Jack	Paul	First Aid	5	300	09:00:00.0000000	12:00:00.0000000	2018-06-14
Jack	Paul	Basket Weaving	6	300	09:00:00.0000000	12:00:00.0000000	2018-06-15
Matt	Joe	Plants	3	200	09:00:00.0000000	12:00:00.0000000	2018-06-11
Matt	Joe	Survival	4	200	09:00:00.0000000	12:00:00.0000000	2018-06-12
Matt	Joe	Insects	5	300	09:00:00.0000000	12:00:00.0000000	2018-06-13
Matt	Joe	Shelter	6	300	09:00:00.0000000	12:00:00.0000000	2018-06-14
Matt	Joe	Reptiles	7	400	09:00:00.0000000	12:00:00.0000000	2018-06-15
Anne	Anderson	Trees	4	200	09:00:00.0000000	12:00:00.0000000	2018-06-11
Anne	Anderson	Animals	5	300	09:00:00.0000000	12:00:00.0000000	2018-06-12
Anne	Anderson	Plants	6	300	09:00:00.0000000	12:00:00.0000000	2018-06-13
Anne	Anderson	Basket Weaving	7	400	09:00:00.0000000	12:00:00.0000000	2018-06-14

Use cases – Report showing a full list of every student's schedule - Continue

- Report will only be executed by administrators and instructors
- Parents and other persons in the system will not have the authorization to run this report
- A critical scenario for this kind of report would be to ensure students are listed in the right camp activities or classes
- At the start of each camp, every member who is administrators or instructors will be eligible to execute this report so that they are aware of the students that are expected to report to a specific camp
- Report will only show students that are stated to be active meaning that their permissions forms, releases forms, and payment have been verified

Use cases – Report showing a full list of every instructor's schedule

Artifact Name: [CAMP.vwTeachersSchedules](#)

Object Type : Views

Description : Reporting the schedule activities/classes for each instructors

Demo :


[SELECT * FROM CAMP.vwTeachersSchedules](#)

FIRST_NAME	LAST_NAME	CLASS_NAME	ROOM_ID	BUILDING_ID	START_TIME	END_TIME	CLASS_DATE
Anuj	Joshi	Animals	1	100	09:00:00.0000000	12:00:00.0000000	2018-06-11
Betty	Pawn	Insects	2	100	09:00:00.0000000	12:00:00.0000000	2018-06-11
Chance	Checker	Plants	3	200	09:00:00.0000000	12:00:00.0000000	2018-06-11
Nate	Chess	Trees	4	200	09:00:00.0000000	12:00:00.0000000	2018-06-11
Maple	Ladder	Survival	5	300	09:00:00.0000000	12:00:00.0000000	2018-06-11
Maple	Ladder	Insects	6	300	13:00:00.0000000	16:00:00.0000000	2018-06-11
Nate	Chess	Plants	7	400	13:00:00.0000000	16:00:00.0000000	2018-06-11
Chance	Checker	Trees	8	400	13:00:00.0000000	16:00:00.0000000	2018-06-11
Betty	Pawn	Survival	9	500	13:00:00.0000000	16:00:00.0000000	2018-06-11
Anuj	Joshi	Animals	10	500	13:00:00.0000000	16:00:00.0000000	2018-06-11
Anuj	Joshi	Plants	2	100	09:00:00.0000000	12:00:00.0000000	2018-06-12
Betty	Pawn	Trees	3	200	09:00:00.0000000	12:00:00.0000000	2018-06-12
Chance	Checker	Survival	4	200	09:00:00.0000000	12:00:00.0000000	2018-06-12
Nate	Chess	Animals	5	300	09:00:00.0000000	12:00:00.0000000	2018-06-12
Maple	Ladder	Insects	6	300	09:00:00.0000000	12:00:00.0000000	2018-06-12
Maple	Ladder	Trees	7	400	13:00:00.0000000	16:00:00.0000000	2018-06-12
Nate	Chess	Survival	8	400	13:00:00.0000000	16:00:00.0000000	2018-06-12
Chance	Checker	Animals	9	500	13:00:00.0000000	16:00:00.0000000	2018-06-12
Betty	Pawn	Insects	10	500	13:00:00.0000000	16:00:00.0000000	2018-06-12
Anuj	Joshi	Plants	1	100	13:00:00.0000000	16:00:00.0000000	2018-06-12
Anuj	Joshi	Survival	3	200	09:00:00.0000000	12:00:00.0000000	2018-06-13
Betty	Pawn	Animals	4	200	09:00:00.0000000	12:00:00.0000000	2018-06-13

Use cases – Report showing a full list of every instructor's schedule - Continue

- Report is available to anyone who is part of the staff
- Administrators will be the primary users to generate this report.
- Report will generate a historical list of classes or activities taught during a specific camp
- Administrators can use this report to adjust or update the schedule as needed for the upcoming summer camp
- Administrators can use this report to make changes due to current staff skills and or because of an unscheduled time off

Use cases – Report showing all the active users

Artifact Name: <u>CAMP.vwCurrentActiveUsers</u>																																								
Object Type : Views																																								
Description : List out all active participants in the camp																																								
Demo : 																																								
<u>SELECT * FROM CAMP.vwCurrentActiveUsers</u>																																								
<table border="1"><thead><tr><th>User name</th><th>ROLE_NAME</th></tr></thead><tbody><tr><td>Sam Mae</td><td>student</td></tr><tr><td>Jack Paul</td><td>student</td></tr><tr><td>Matt Joe</td><td>student</td></tr><tr><td>Anne Anderson</td><td>student</td></tr><tr><td>Paul Smith</td><td>student</td></tr><tr><td>Frank Lynn</td><td>student</td></tr><tr><td>Hillary Clinton</td><td>student</td></tr><tr><td>Joe Coffee</td><td>student</td></tr><tr><td>Gladis Night</td><td>student</td></tr><tr><td>Clay Walker</td><td>student</td></tr><tr><td>Sal Mae</td><td>student</td></tr><tr><td>Jeff Mae</td><td>parent</td></tr><tr><td>Rand Paul</td><td>parent</td></tr><tr><td>Cool Joe</td><td>parent</td></tr><tr><td>Mike Anderson</td><td>parent</td></tr><tr><td>Will Smith</td><td>parent</td></tr><tr><td>Sue Lynn</td><td>parent</td></tr><tr><td>Bernie Trump</td><td>parent</td></tr><tr><td>Chuck Coffee</td><td>parent</td></tr></tbody></table>	User name	ROLE_NAME	Sam Mae	student	Jack Paul	student	Matt Joe	student	Anne Anderson	student	Paul Smith	student	Frank Lynn	student	Hillary Clinton	student	Joe Coffee	student	Gladis Night	student	Clay Walker	student	Sal Mae	student	Jeff Mae	parent	Rand Paul	parent	Cool Joe	parent	Mike Anderson	parent	Will Smith	parent	Sue Lynn	parent	Bernie Trump	parent	Chuck Coffee	parent
User name	ROLE_NAME																																							
Sam Mae	student																																							
Jack Paul	student																																							
Matt Joe	student																																							
Anne Anderson	student																																							
Paul Smith	student																																							
Frank Lynn	student																																							
Hillary Clinton	student																																							
Joe Coffee	student																																							
Gladis Night	student																																							
Clay Walker	student																																							
Sal Mae	student																																							
Jeff Mae	parent																																							
Rand Paul	parent																																							
Cool Joe	parent																																							
Mike Anderson	parent																																							
Will Smith	parent																																							
Sue Lynn	parent																																							
Bernie Trump	parent																																							
Chuck Coffee	parent																																							

Use cases – Report showing all the active users - continue

- Report allows administrators to view active participants in the summer camp.
- Camp administrators can send notification and newsletters based on this report
- Security crew members will have a list of the number of participants are expected to be allowed in the camp
- Applications built on top of this database can use this list to check on the availability of a user to view specific reports

Use cases – Report showing all the inactive users

Artifact Name: <u>CAMP.vwCurrentInactiveUsers</u>	
Object Type : Views	
Description : List out all inactive participants in the camp	
Demo :	
<pre>SELECT * FROM <u>CAMP.vwCurrentInactiveUsers</u></pre>	
User name	ROLE_NAME
Mort Knight	administrator
Walter Rook	work crews
Susan Castle	instructor
Noah Kelly	student
Emma Mason	student
Olivia Brock	student
Jacob Kent	student
Ava Lester	student
Ethan Jones	student
Mia Tea	student
Alex Light	student
Sophia Mills	student
Jackson Kelly	parent
Matt Mason	parent
Dan Brock	parent
Logan Kent	parent
Harper Lester	parent
Lucas Jones	parent
Avery Tea	parent
Ella Light	parent
Eva Mills	parent

Use cases – Report showing all the inactive users - Continue

- Report allows administrators to view inactive participants in the summer camp.
- Security crew members will have a list of participants that are not allowed to be in camp buildings or classrooms
- Applications built on top of this database can use this list to check on the availability of a user to view specific reports

Use cases – Report to check a student's permission

Artifact Name: CAMP.usp_getstudentpermissions

Object Type : Stored Procedure

Description : Provides student's permissions and it's start and end date

Demo :

```
DECLARE @userId int  
SET @userId = 3  
EXECUTE CAMP.[usp_getstudentpermissions] @userId
```

	FIRST_NAME	LAST_NAME	PERMISSION_START_DATE	PERMISSION_END_DATE	PERMISSION_TYPE	PERMISSION_TYPE_DESC
1	Matt	Joe	2018-06-11	2018-06-15	Off Camp	permission to go off camp areas

Use cases – Report to check a student's permission - Continue

- Verify a student's permissions, for example, if a student has permission to go off camp areas, then it will report it
- Administrators, students, and parents can execute this stored procedures
- Provide a duration for the exemption

Use cases – Report to display all student's permissions

Artifact Name: CAMP.vw_getstudentpermissions

Object Type : View

Description : Provides student's permissions and it's start and end date

Demo :

```
SELECT * FROM CAMP.[vw_getstudentpermissions]
```

	FIRST_NAME	LAST_NAME	PERMISSION_START_DATE	PERMISSION_END_DATE	PERMISSION_TYPE	PERMISSION_TYPE_DESC
1	Sam	Mae	2018-06-11	2018-06-13	Leave Early	permission to leave camp early
2	Sam	Mae	2018-06-11	2018-06-13	Drive	permission to drive to and from camp
3	Sam	Mae	2018-06-11	2018-06-15	Media Share	permission to share photos on social media
4	Sam	Mae	2018-06-11	2018-06-15	ALT Parents	permission to leave camp with other parents
5	Sam	Mae	2018-06-11	2018-06-15	Off Camp	permission to go off camp areas
6	Jack	Paul	2018-06-11	2018-06-15	Off Camp	permission to go off camp areas
7	Matt	Joe	2018-06-11	2018-06-15	Off Camp	permission to go off camp areas
8	Anne	Anderson	2018-06-11	2018-06-15	Off Camp	permission to go off camp areas
9	Paul	Smith	2018-06-11	2018-06-15	Off Camp	permission to go off camp areas

Use cases – **Report to display all students' release information - Continue**

- Report to provide release information of all the students
- Provide details about release type and its description with release start and end date.
- Administrators and instructors have access to this report

Use cases – Report to display a student's release information

Artifact Name: CAMP.usp_getreleaseinformation						
Object Type : Stored Procedure						
Description : Provides release information of a student						
Demo :						
<pre>DECLARE @userId int SET @userId = 4 EXECUTE CAMP.[usp_getreleaseinformation] @userId</pre>						
	FIRST_NAME	LAST_NAME	RELEASE_START_DATE	RELEASE_END_DATE	RELEASE_TYPE	RELEASE_TYPE_DESC
1	Anne	Anderson	2018-06-11	2018-06-15	Non-consent	non-consent to release

Use cases – Report to display a student's release information - Continue

- Allow student to view their release info
- Allow parents to confirm that their children have the correct release information
- Allow administrators and instructors to get release information for a student quickly

Use cases – Report to display all student's release information

Artifact Name: CAMP.vw_getreleaseinformation

Object Type : View

Description : Provides release information of a student

Demo :

```
SELECT * FROM CAMP.[vw_getreleaseinformation]
```

	FIRST_NAME	LAST_NAME	RELEASE_START_DATE	RELEASE_END_DATE	RELEASE_TYPE	RELEASE_TYPE_DESC
1	Sam	Mae	2018-06-11	2018-06-15	Exception	exception to release
2	Jack	Paul	2018-06-11	2018-06-15	Consent	consent to release
3	Matt	Joe	2018-06-11	2018-06-15	Consent	consent to release
4	Anne	Anderson	2018-06-11	2018-06-15	Non-consent	non-consent to release
5	Paul	Smith	2018-06-11	2018-06-15	Consent	consent to release
6	Frank	Lynn	2018-06-11	2018-06-15	Consent	consent to release
7	Hillary	Clinton	2018-06-11	2018-06-15	Consent	consent to release
8	Joe	Coffee	2018-06-11	2018-06-15	Consent	consent to release
9	Gladis	Night	2018-06-11	2018-06-15	Consent	consent to release
10	Clay	Walker	2018-06-11	2018-06-15	Consent	consent to release
11	Sal	Mae	2018-06-11	2018-06-15	Exception	exception to release

Use cases – Report to display all student's release information - Continue

- Allow administrators and instructors to get release information for a student quickly
- Provide security personnel a list of all the students release detailed information

Use cases – Report to display all student's payment information

Artifact Name: CAMP.vw_getpaymentinformation

Object Type : View

Description : Provides payment information of students

Demo :

```
SELECT * FROM CAMP.[vw_getpaymentinformation]
```

	FIRST_NAME	LAST_NAME	PAYMENT_DATE	PAYMENT_DUE_DATE	PAYMENT_STATUS	CAMP_YEAR
1	Sam	Mae	2017-01-11	2017-12-01	paid	2018
2	Jack	Paul	2017-03-21	2017-12-01	paid	2018
3	Matt	Joe	2017-09-13	2017-12-01	paid	2018
4	Anne	Anderson	2017-07-18	2017-12-01	paid	2018
5	Paul	Smith	2017-02-19	2017-12-01	paid	2018
6	Frank	Lynn	2017-08-22	2017-12-01	paid	2018
7	Hillary	Clinton	2017-11-19	2017-12-01	paid	2018
8	Joe	Coffee	2017-06-09	2017-12-01	paid	2018
9	Gladis	Night	NULL	2017-12-01	not paid	2018
10	Clay	Walker	NULL	2017-12-01	not paid	2018
11	Sal	Mae	2017-07-01	2017-12-01	paid	2018

Use cases – **Report to display all student's payment information - Continue**

- Provide administrators a list of the payment information of students
- Provide details about payment due date, payment status, and payment date.
- Instructors, parents, and students are not allowed to view this report

Use cases – Report to display a student's payment information

Artifact Name: CAMP.usp_getpaymentinformation

Object Type : Stored Procedure

Description : Provides payment information of a student

Demo :

```
DECLARE @userId int  
SET @userId = 4  
EXECUTE CAMP.[usp_getpaymentinformation] @userId
```

	FIRST_NAME	LAST_NAME	PAYMENT_DATE	PAYMENT_DUE_DATE	PAYMENT_STATUS	CAMP_YEAR
1	Anne	Anderson	2017-07-18	2017-12-01	paid	2018

Use cases – **Report to display a student's payment information - Continue**

- Parents can generate this stored proc to get payment information
- Students and instructors are not allowed to run this stored procedure
- Administrators can use this stored procedure to check on a student's payment status quickly

Use cases – Report to display a student's detailed information

Artifact Name: CAMP.usp_GetChildDetails

Object Type : Stored Procedure

Description : Provides required information about students to their parents

Demo :

```
DECLARE @userId int  
SET @userId = 22  
EXECUTE CAMP.[usp_GetChildDetails] @userId
```

	FIRST_NAME	LAST_NAME	DATE_OF_BIRTH	PAYMENT_DATE	PAYMENT_DUE_DATE	PAYMENT_STATUS	CAMP_YEAR
1	Jack	Paul	2007-02-02	2017-11-21	2017-12-01	paid	2018

Use cases – **Report to display a student's detailed information - Continue**

- Parent's will execute this report to confirm that their children data is correct
- Administrators will perform this report to get information to celebrate student's birthdays, or to organize students based on age

Use cases – Report to display camp's weekly activities

Artifact Name: CAMP.vw_getweeklyschedule

Object Type : View

Description : Provides weekly scheduled of all the activities in the camp

Demo :

SELECT * FROM CAMP.[vw_getweeklyschedule]

	CLASS_NAME	ROOM_NAME	BUILDING_ID	SCHEDULE_ID	CLASS_DATE	START_TIME	END_TIME
1	Animals	Red A	100	51111	2017-06-12	09:00:00.0000000	12:00:00.0000000
2	Insects	Red B	100	51112	2017-06-12	09:00:00.0000000	12:00:00.0000000
3	Plants	White A	200	51113	2017-06-12	09:00:00.0000000	12:00:00.0000000
4	Trees	White B	200	51114	2017-06-12	09:00:00.0000000	12:00:00.0000000
5	Survival	Green A	300	51115	2017-06-12	09:00:00.0000000	12:00:00.0000000
6	Insects	Green B	300	51121	2017-06-12	13:00:00.0000000	16:00:00.0000000
7	Plants	Blue A	400	51122	2017-06-12	13:00:00.0000000	16:00:00.0000000
8	Trees	Blue B	400	51123	2017-06-12	13:00:00.0000000	16:00:00.0000000
9	Survival	Pink A	500	51124	2017-06-12	13:00:00.0000000	16:00:00.0000000
10	Animals	Pink B	500	51125	2017-06-12	13:00:00.0000000	16:00:00.0000000
11	Plants	Red B	100	51211	2017-06-13	09:00:00.0000000	12:00:00.0000000
12	Trees	White A	200	51212	2017-06-13	09:00:00.0000000	12:00:00.0000000
13	Survival	White B	200	51213	2017-06-13	09:00:00.0000000	12:00:00.0000000
14	Animals	Green A	300	51214	2017-06-13	09:00:00.0000000	12:00:00.0000000
15	Insects	Green B	300	51215	2017-06-13	09:00:00.0000000	12:00:00.0000000
16	Trees	Blue A	400	51221	2017-06-13	13:00:00.0000000	16:00:00.0000000
17	Survival	Blue B	400	51222	2017-06-13	13:00:00.0000000	16:00:00.0000000
18	Animals	Pink A	500	51223	2017-06-13	13:00:00.0000000	16:00:00.0000000
19	Insects	Pink B	500	51224	2017-06-13	13:00:00.0000000	16:00:00.0000000

Use cases – **Report to display camp's weekly activities**

- Report provides a list of all the activities that are happening during camp
- Report contains the locations where the events are taken place
- Report provides the time that the events will be taken place
- Administrators, students, parents, and instructors will have access to this report

Use cases – Report to display where a specific event is taken place


Artifact Name: CAMP.usp_GetCurrentBuildingActivity

Object Type : Stored Procedure

Description : This procedure will fetch current list of activities for a building, including rooms, classes and total an estimated #students in the class.

Demo :

EXEC [CAMP].[usp_GetCurrentBuildingActivity] 'Red Barn'

 **EXEC** [CAMP].[usp_GetCurrentBuildingActivity] 'Red Barn'|

10 %

Results **Messages**

	CLASS_NAME	STAFF_NAME	ROOM_NAME	TOTAL_STUDENTS
1	Animals	AnujJoshi	Red A	3
2	Insects	BettyPawn	Red B	2
3	Plants	AnujJoshi	Red B	3
4	Plants	AnujJoshi	Red A	2
5	Shelter	BettyPawn	Red A	2
5	Basket Weaving	AnujJoshi	Red B	2
7	Basket Weaving	ChanceChecker	Red A	2
3	Reptiles	BettyPawn	Red B	2
3	Reptiles	NateChess	Red A	2
10	Canoe	ChanceChecker	Red B	2
11	Animals	SusanCastle	Red A	3

Use cases – **Report to display where a specific event is taken place - Continue**

- Administrators, and instructors will execute this report when trying to find where an event is taken place quickly
- Security personnel will use this report in case of emergency
- Parents and students will also have access to this report to find out where their classes are taken place

Use cases – Report to display what roles are assigned to each users

Artifact Name: CAMP.vwUserRoles

Object Type : View

Description : This is an all up View to view all users and the roles they are assigned to.

Demo :

SELECT * FROM CAMP.[vwUserRoles]

100 %

Results Messages

	USER_ID	FULL_NAME	IS_ADMIN	IS_INSTRUCTOR	IS_STUDENT	IS_PARENT	IS_WORKCREW	USER_STATUS
1	1	Sam Mae	No	No	Yes	No	No	Active
2	2	Jack Paul	No	No	Yes	No	No	Active
3	3	Matt Joe	No	No	Yes	No	No	Active
4	4	Anne Anderson	No	No	Yes	No	No	Active
5	5	Paul Smith	No	No	Yes	No	No	Active
6	6	Frank Lynn	No	No	Yes	No	No	Active
7	7	Hillary Clinton	No	No	Yes	No	No	Active
8	8	Joe Coffee	No	No	Yes	No	No	Active
9	9	Gladis Night	No	No	Yes	No	No	Active
10	10	Clay Walker	No	No	Yes	No	No	Active
11	11	Sol Mae	No	No	Yes	No	No	Active
12	21	Jeff Mae	No	No	No	Yes	No	Active
13	22	Rand Paul	No	No	No	Yes	No	Active
14	23	Cool Joe	No	No	No	Yes	No	Active
15	24	Mike Anderson	No	No	No	Yes	No	Active
16	25	Will Smith	No	No	No	Yes	No	Active
17	26	Sue Lynn	No	No	No	Yes	No	Active
18	27	Bemie Trump	No	No	No	Yes	No	Active
19	28	Chuck Coffee	No	No	No	Yes	No	Active
20	29	Fred Night	No	No	No	Yes	No	Active
21	30	Mary Walker	No	No	No	Yes	No	Active
22	41	Jefferson King	Yes	No	No	No	No	Active
23	42	Ray Queen	Yes	No	No	No	No	Active
24	43	Cory Bishop	Yes	No	No	No	No	Active
25	44	Mort Knight	Yes	No	No	No	No	InActive
26	45	Walter Rook	No	No	No	No	Yes	InActive
27	51	Susan Castle	No	Yes	No	No	No	InActive
28	52	Betty Pawn	No	Yes	No	No	No	Active
29	53	Chance Chec...	No	Yes	No	No	No	Active

Use cases – **Report to display what roles are assigned to each user - Continue**

- Report is only available to administrators
- Report provides critical information on roles and authorizations available to each user who is active in the system
- Report can be shared with security personnel in case of potential hacking

Administrator cases – Report to show what permissions are permitted for a student

Artifact Name: CAMP.fn_Check_Permission

Object Type : Function

Description :

This Function will enable users to verify a permission for a student.

Demo :

```
SELECT CAMP.fn_Check_Permission(1, 'drive') AS IsPermitted  
SELECT CAMP.fn_Check_Permission(1, 'Leave Early') AS IsPermitted  
SELECT CAMP.fn_Check_Permission(2, 'Leave Early') AS IsPermitted
```

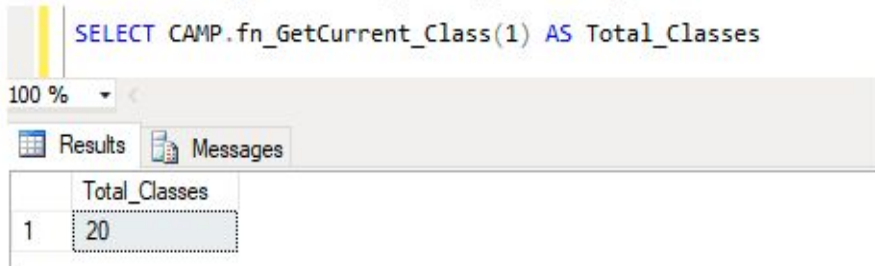
```
SELECT CAMP.fn_Check_Permission(1, 'drive') AS IsPermitted  
SELECT CAMP.fn_Check_Permission(1, 'Leave Early') AS IsPermitted  
SELECT CAMP.fn_Check_Permission(2, 'Leave Early') AS IsPermitted
```

0 %
Results Messages
IsPermitted
Yes
IsPermitted
No
IsPermitted
No

Administrator cases – **Report to show what permissions are permitted for a student**

- Report allows instructors and administrators to check on a student's permission
- A student's permissions can be removed based on whether this report is accurate

Administrator cases – Report to show number of classes a student is/has enrolled

Artifact Name: CAMP.fn_GetCurrent_Class		
Object Type : Function		
Description : This Function will list the total classes the student is enrolled/scheduled for.		
Demo : SELECT CAMP.fn_GetCurrent_Class(1) AS Total_Classes  <p>The screenshot shows a SQL query execution interface. The query is <code>SELECT CAMP.fn_GetCurrent_Class(1) AS Total_Classes</code>. Below the query, there is a 'Results' tab and a 'Messages' tab. The 'Results' tab is active, showing a table with one column 'Total_Classes' and one row with the value '20'.</p> <table><thead><tr><th>Total_Classes</th></tr></thead><tbody><tr><td>20</td></tr></tbody></table>	Total_Classes	20
Total_Classes		
20		

Administrator cases – **Report showing the number of classes a student is enrolled**

- Administrators use this report to find out the number of classes a student has taken throughout his/her time attending the camp
- Report shows effectiveness of the camp based on total classes attended in a lifetime
-

Administrator cases – Trigger when updating payment due date

Artifact Name: STUDENT_PAYMENT_UPDATE

Object Type : Trigger

Description : A warning message is thrown if anyone is trying to change a camp's payment due date

Demo :

```
USE SUMMER_CAMP
UPDATE [SUMMER_CAMP].[CAMP].[MAP_STUDENT_PAYMENT]
  SET PAYMENT_DUE_DATE = CAST('2017-12-01' AS DATE)
WHERE CAMP_YEAR = '2018'
```

Updating the payment due date for a specific camp is not allowed
Msg 3609, Level 16, State 1, Line 120
The transaction ended in the trigger. The batch has been aborted.

Administrator cases – **Trigger when updating payment due date - Continue**

- Trigger protects the system from camp's payment due date.
- Administrators will set the original payment due date, and this date is final
- Protect parents from any unexpected events of payment due date being changed

Administrator cases – Trigger when deleting a user

Artifact Name: USER_DELETE

Object Type : Trigger

Description : A warning message is thrown if anyone is trying to remove any users from the user table

Demo :


```
USE SUMMER_CAMP
DELETE FROM [SUMMER_CAMP].[CAMP].[MAP_USER_ROLE]
WHERE USER_ID = 45
GO
DELETE FROM [CAMP].[STAFF]
WHERE STAFF_ID = 45
GO
DELETE FROM [SUMMER_CAMP].[CAMP].[USER]
WHERE USER_ID = 45
```

Msg 50000, Level 16, State 1, Procedure USER_DELETE, Line 6 [Batch Start Line 149]
You cannot delete any users instead set the Is_ACTIVE flag to false for the related user
Msg 3609, Level 16, State 1, Line 150
The transaction ended in the trigger. The batch has been aborted.

Administrator cases – **Trigger when deleting a user**

- User table is meant to retain users information for life.
- This trigger is intended to ensure that no users will ever be deleted.
- If a user is no longer an active participant in the camp, then the best option is to set the `is_active` flag to false.
- Security layer if anyone were to hack the system and to try to manipulate the data

Administrator cases – Function to check what students have more than one permissions

Artifact Name: fx_StudentWithMultiPermissions					
Object Type : Function					
Description : returns the name of students who have more than one permissions					
Demo :					
select * from fx_StudentWithMultiPermissions()					
<table border="1"><thead><tr><th>StudentName</th><th>Total</th></tr></thead><tbody><tr><td>Sam Mae</td><td>5</td></tr></tbody></table>	StudentName	Total	Sam Mae	5	
StudentName	Total				
Sam Mae	5				

Administrator cases – **Function to check what students have more than one permissions - Continue**

- Enable staff to know which students have more than one permissions
- List to show the students that will require extra monitoring because of the number of permissions
- Only administrators will be allowed to run this function

Additional Enhancements

- Future enhancements
 - Optionally list summer camp activities that can happen outdoor
 - Allow parents to make changes associated with their contact info
 - Add an entry for meal planning for those who require special diet
 - Allow students to update their meal preferences
 - Provide parents a way to remove their child from a camp program through the application
 - Provide pick up and drop off transportation schedule for those who cannot drive to the campground
 - Provide an option to have partial payments from divorced or separated parents

Project Dynamics

- Robert Marshal
 - Created database objects and added test data
- Ram Ganesan
 - Reviewed and redesigned original database objects, and created stored procedures, functions, and triggers
- Anuj Joshi
 - Created stored procedures, and views
- Pierre Augustamar
 - Handled all the documentation, and created stored procedures, functions, and triggers

Project Dynamics - Continue

- We used an agile software development with rapid development.
- With constant changes to the design after a team member raises a concern, then an agile methodology was best suited for this sort of group project
- We continuously review the requirements to make sure that we are compliant with what was being asked
- We had team meetings every other day at the beginning of the project; then we had sync up sessions every day during the last week that the project was due.
- We had an open discussion style where everyone can voice his opinions
- We had a democratic system where everyone's inputs are taken into consideration before making any actions
- All design changes required approval from everyone

Lessons Learned

- Review the original design during the first group meeting so that we can avoid multiple iterations later on
- Have concrete deliverables early in the project
- Utilize teammate's strength early in the project

QUESTIONS