**MIS 6326.001 DATA MANAGEMENT**

**PROJECT APP – WUNDERLIST**

Team members

AKSHAY ALSE

NEERAJKUMAR PATHAK

PRATEEK RAWAT

SYED RIAZUDEEN SHAHABUDEEN

SRIRAM PRASATH S PANDIARAJAN

**Wunderlist Introduction**

**Wunderlist**is a cloud-based [task management](https://en.wikipedia.org/wiki/Task_management) application. It allows users to manage their tasks from a smartphone, tablet, computer and [smartwatch](https://en.wikipedia.org/wiki/Smartwatch). Wunderlist is free; additional collaboration features are available in a paid version known as Wunderlist Pro (released April 2013). Wunderlist was created in 2011 by [Berlin](https://en.wikipedia.org/wiki/Berlin)-based start-up 6Wunderkinder.

**Project Introduction**

In this project, we have considered different features/aspects available in Wunderlist. Accordingly, an ERD is designed. A sample DB has been created with entities in ERD as tables and sample data has been inserted. Using this data, we have formed scenarios which will help us understand SQL in practical conditions.

**Narrative**

USER can create new Wunderlist account using existing Facebook, Windows, Gmail account or can register by giving Name, Email-id and password. A confirmation mail will be sent after registering for the app. User can create zero or many list. A LIST out of Groceries, Private, Travel, Movies to watch, Work, Family or can create a self-defined custom list. List can have many tasks. List can be shared by many users included in a task

All the task can be assigned to users in a list and it includes date reminders. Conversations can be made related to task. Each task can have one conversation thread. Activity can be assigned to one or more users.

**Additional feature:**

**Export:**  As an additional feature, an option to export the conversation history related to the task and activity is being added. Using this feature, we can keep record of conversations made for all task and activity.

**ERD Diagram:**



**ERD Digaram With Additional Feature:**



**Explain Entities and Attributes**

|  |  |
| --- | --- |
| **Entities** | **Attributes** |
| **USER**  Users having an account in wunderlist app. | **User\_id**  First\_name  Last\_name  Email\_id |
| **LIST**  Lists created by users | **List\_id**  List\_name |
| **USER\_LIST**  As relationship between User and List happens to be a many to many relationship, we break the relationship to form this new entity. | **User\_id**  **List\_id** |
| **CONVERSATION**  Conversations can be initiated in the app related to tasks and activities. | **Conversation\_id**  **Task\_id**  **List\_id**  Chat\_time |
| **TASK**  A list can have tasks created by users. Task completion date has to be set while creating the task. | **List\_id**  **Task\_id**  User\_id  Date\_reminder  Task\_name |
| **ACTIVITY**  Task created can be assigned to any user by any user in any list. Activities done with the task are recorded in this entity | **Activity\_id**  User\_id(Assign\_to)  User\_id(Assign\_by)  List\_name  Task\_id  Assigned\_date  Completed\_date |
| **EXPORT**  An option to export the conversation history related to the task and activity is being added. Using this feature, we can keep record of conversations made for all task and activity. | **Export\_id**  **Conversation\_id**  Email\_id |

**Tables:**

**Table structure for table `user`**

CREATE TABLE `user` (

`user\_id` varchar(6) NOT NULL,

`first\_name` varchar(20) NOT NULL,

`last\_name` varchar(20) NOT NULL,

`email\_id` varchar(50) NOT NULL

)

**Indexes for table `user`**

ALTER TABLE `user`

ADD PRIMARY KEY (`user\_id`);

**Dumping data for table `user`**

INSERT INTO `user` (`user\_id`, `first\_name`, `last\_name`, `email\_id`) VALUES

('10001', 'pooja', 'kanchan', 'pxk171130@utdallas.edu'),

('10002', 'prateek', 'rawat', 'prr170130@utdallas.edu'),

('10003', 'akshay', 'alse', 'ava170330@utdallas.edu'),

('10004', 'neeraj', 'pathak', 'nrp171130@utdallas.edu'),

('10005', 'suresh', 'bhusare', 'ssb171130@utdallas.edu'),

('10006', 'abhiraj', 'trump', 'asb71130@utdallas.edu'),

('10007', 'jasmeet', 'harjai', 'jsh171130@utdallas.edu');

**Table structure for table `list`**

CREATE TABLE `list` (

`list\_id` varchar(6) NOT NULL,

`list\_name` varchar(15) NOT NULL

)

**Dumping data for table `list`**

INSERT INTO `list` (`list\_id`, `list\_name`) VALUES

('1a', 'Inbox'),

('1b', 'Project'),

('1c', 'Project'),

('2a', 'Project'),

('2b', 'Family'),

('2c', 'Movies'),

('3a', 'Travel'),

('3b', 'Movies'),

('3c', 'Private'),

('4a', 'Movies');

**Indexes for table `list`**

ALTER TABLE `list`

ADD PRIMARY KEY (`list\_id`);

**Table structure for table `task`**

CREATE TABLE `task` (

`task\_id` varchar(6) NOT NULL,

`task\_name` varchar(15) NOT NULL,

`user\_id` varchar(6) NOT NULL,

`list\_id` varchar(6) NOT NULL,

`date\_reminder` datetime NOT NULL

)

**Dumping data for table `task`**

INSERT INTO `task` (`task\_id`, `task\_name`, `user\_id`, `list\_id`, `date\_reminder`) VALUES

('t1', 'find\_app', '10002', '1b', '2017-11-30 23:59:00'),

('t10', 'internship\_call', '10001', '1a', '2017-11-22 00:00:00'),

('t11', 'Create\_Report', '10001', '1c', '2017-11-30 00:00:00'),

('t2', 'review', '10005', '1b', '2017-12-01 00:00:00'),

('t3', 'define\_narrativ', '10007', '1b', '2017-11-28 00:00:00'),

('t4', 'thanksgiving', '10001', '2b', '2017-11-23 00:00:00'),

('t5', 'padmavati', '10003', '4a', '2017-12-30 00:00:00'),

('t6', 'DBW', '10006', '1c', '2017-11-29 00:00:00'),

('t7', 'christmas', '10002', '2b', '2017-12-25 00:00:00'),

('t8', 'newyork', '10005', '3a', '2017-12-26 00:00:00'),

('t9', 'laundry', '10004', '3c', '2017-11-22 00:00:00');

**Indexes for table `task`**

ALTER TABLE `task`

ADD PRIMARY KEY (`task\_id`,`list\_id`),

ADD KEY `list\_id` (`list\_id`);

**Constraints for table `task`**

ALTER TABLE `task`

ADD CONSTRAINT `task\_ibfk\_1` FOREIGN KEY (`list\_id`) REFERENCES `list` (`list\_id`);

**Table structure for table `user\_list`**

CREATE TABLE `user\_list` (

`user\_id` varchar(6) NOT NULL,

`list\_id` varchar(6) NOT NULL

)

**Dumping data for table `user\_list`**

INSERT INTO `user\_list` (`user\_id`, `list\_id`) VALUES

('10001', '1a'),

('10002', '1b'),

('10003', '1c'),

('10004', '2a'),

('10005', '2b'),

('10006', '2c'),

('10007', '3a');

**Indexes for table `user\_list`**

ALTER TABLE `user\_list`

ADD PRIMARY KEY (`user\_id`,`list\_id`),

ADD KEY `list\_id` (`list\_id`);

**Constraints for table `user\_list`**

ALTER TABLE `user\_list`

ADD CONSTRAINT `user\_list\_ibfk\_1` FOREIGN KEY (`list\_id`) REFERENCES `list` (`list\_id`),

ADD CONSTRAINT `user\_list\_ibfk\_2` FOREIGN KEY (`user\_id`) REFERENCES `user` (`user\_id`) ON DELETE CASCADE ON UPDATE CASCADE;

COMMIT;

**Table structure for table `activity`**

CREATE TABLE `activity` (

`activity\_id` varchar(6) NOT NULL,

`assign\_to` varchar(6) NOT NULL,

`assign\_by` varchar(6) NOT NULL,

`task\_id` varchar(6) NOT NULL,

`assigned\_date` date NOT NULL,

`completed\_date` date NOT NULL

)

**Dumping data for table `activity`**

INSERT INTO `activity` (`activity\_id`, `assign\_to`, `assign\_by`, `task\_id`, `assigned\_date`, `completed\_date`) VALUES

('a1', '10002', '10002', 't1', '2017-10-30', '2017-11-29'),

('a2', '10001', '10003', 't4', '2017-11-30', '2017-12-22'),

('a3', '10007', '10007', 't3', '2017-11-15', '2017-11-28'),

('a4', '10003', '10002', 't5', '2017-11-30', '2017-12-26'),

('a5', '10002', '10001', 't7', '2017-11-30', '2017-12-12'),

('a6', '10004', '10001', 't9', '2017-10-30', '2017-11-22');

**Indexes for table `activity`**

ALTER TABLE `activity`

ADD PRIMARY KEY (`activity\_id`),

ADD KEY `assign\_to` (`assign\_to`),

ADD KEY `assign\_by` (`assign\_by`);

**Constraints for table `activity`**

ALTER TABLE `activity`

ADD CONSTRAINT `activity\_ibfk\_1` FOREIGN KEY (`assign\_to`) REFERENCES `user` (`user\_id`),

ADD CONSTRAINT `activity\_ibfk\_2` FOREIGN KEY (`assign\_by`) REFERENCES `user` (`user\_id`);

**Table structure for table `coversation`**

CREATE TABLE `coversation` (

`converstion\_id` varchar(6) NOT NULL,

`chat\_time` datetime NOT NULL,

`task\_id` varchar(6) NOT NULL,

`list\_id` varchar(6) NOT NULL

)

**Dumping data for table `coversation`**

INSERT INTO `coversation` (`converstion\_id`, `chat\_time`, `task\_id`, `list\_id`) VALUES

('c1', '2017-11-28 00:00:00', 't2', '1b'),

('c2', '2017-11-20 00:00:00', 't4', '2b'),

('c3', '2017-12-25 00:00:00', 't5', '4a'),

('c4', '2017-12-10 00:00:00', 't5', '4a'),

('c5', '2017-11-20 00:00:00', 't6', '1c'),

('c6', '2017-11-17 00:00:00', 't9', '3c');

**Indexes for table `coversation`**

ALTER TABLE `coversation`

ADD PRIMARY KEY (`converstion\_id`,`task\_id`),

ADD KEY `task\_id` (`task\_id`),

ADD KEY `coversation\_ibfk\_2` (`list\_id`);

**Constraints for table `coversation`**

ALTER TABLE `coversation`

ADD CONSTRAINT `coversation\_ibfk\_1` FOREIGN KEY (`task\_id`) REFERENCES `task` (`task\_id`),

ADD CONSTRAINT `coversation\_ibfk\_2` FOREIGN KEY (`list\_id`) REFERENCES `task` (`list\_id`);

**Table structure for table `export`**

CREATE TABLE `export` (

`export\_id` varchar(6) NOT NULL,

`converstion\_id` varchar(6) NOT NULL,

`email\_id` varchar(50) NOT NULL

)

**Dumping data for table `export`**

INSERT INTO `export` (`export\_id`, `converstion\_id`, `email\_id`) VALUES

('e1', 'c2', 'meetprateekrawat@gmail.com'),

('e2', 'c5', 'pxk171130@utdallas.edu'),

('e3', 'c1', 'ssb171130@utdallas.edu'),

('e4', 'c1', 'jsh171130@utdallas.edu'),

('e5', 'c4', 'asb71130@utdallas.edu'),

('e6', 'c3', 'nrp171130@utdallas.edu');

**Indexes for table `export`**

ALTER TABLE `export`

ADD PRIMARY KEY (`export\_id`,`converstion\_id`),

ADD KEY `converstion\_id` (`converstion\_id`);

**Constraints for table `export`**

ALTER TABLE `export`

ADD CONSTRAINT `export\_ibfk\_1` FOREIGN KEY (`converstion\_id`) REFERENCES `coversation` (`converstion\_id`);

**SCENARIOS**

**Scenario 1:**

1. **Select first name, last name of user where date of completion(date\_reminder) is less than today’s date.**

**(It is useful for user to track the due date of the task associated with the list, it gives advantage to user to plan his/ her activities accordingly.)**

*SELECT*

*first\_name, last\_name*

*FROM*

*USER u*

*JOIN user\_list ul ON*

*u.user\_id = ul.user\_id*

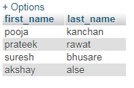
*JOIN task t ON*

*ul.list\_id = t.list\_id*

*WHERE*

*t.date\_reminder < CURRENT\_DATE*

**RESULT:**



1. **Select userid, first name, last name,list name where completed date is equal to current date.  
   (This helps the user to track the completed tasks in the list, this can be useful for validating the completed tasks before recreating the same sort of task.)**

*SELECT*

*u.user\_id, u.first\_name, u.last\_name, t.task\_name*

*FROM*

*USER u*

*JOIN task t ON*

*u.user\_id = t.user\_id*

*JOIN activity a ON*

*a.task\_id = t.task\_id*

*WHERE*

*a.completed\_date = CURRENT\_DATE.*

**RESULT:**



1. **Select first name,last name of user who has export id equivalent to email id :-** [**meetprateekrawat@gmail.com**](mailto:meetprateekrawat@gmail.com)**.**

**(It is useful to save the conversation and mail it to the respective email-id as per user’s convenience.)**

*SELECT*

*u.user\_id, u.first\_name, u.last\_name*

*FROM*

*USER u*

*JOIN task t ON*

*t.user\_id = u.user\_id*

*JOIN coversation c ON*

*c.task\_id = t.task\_id*

*JOIN EXPORT e ON*

*e.converstion\_id = c.converstion\_id*

*WHERE*

*e.email\_id = 'meetprateekrawat@gmail.com'*

**RESULT:**



1. **Select first name, last name of user who has been assigned maximum number of tasks.**

**(Using this user can track who has been assigned maximum work in a task.)**

*SELECT*

*u.first\_name, u.last\_name, COUNT(\*)*

*FROM*

*USER u*

*JOIN task t ON*

*u.user\_id = t.user\_id*

*GROUP BY*

*u.user\_id*

*ORDER BY*

*COUNT(\*)*

*DESC*

*LIMIT 1*

**RESULT:**



1. **Select conversation id which belongs to list name ‘movies’.**

**(Using this you may come to know what kind of movies people prefer to watch, respecting the privacy of the user off course)**

*SELECT*

*converstion\_id*

*FROM*

*coversation c*

*WHERE*

*c.list\_id IN(*

*SELECT*

*list\_id*

*FROM LIST*

*l*

*WHERE*

*l.list\_name = 'Movies'*

*)*

**RESULT:**



6.**Provide list\_name and associated task with user name, assigned to a user by himself/ herself.**

**(Such tasks most likely to be performed by individual and not by group, helps to know such tasks)**

*SELECT*

*u.first\_name, u.last\_name, t.task\_name, l.list\_name*

*FROM*

*activity a*

*JOIN USER u ON*

*a.assign\_to = u.user\_id*

*JOIN task t ON*

*a.task\_id = t.task\_id*

*JOIN LIST l ON*

*l.list\_id = t.list\_id*

*WHERE*

*a.assign\_to = a.assign\_by*

**RESULT:**



**7.Select all the list names which have been assigned to users.**

**(It is useful to know how many tasks are assigned to each user in general, help to know how frequently app being used by users.)**

*SELECT*

*u.first\_name, u.last\_name, GROUP\_CONCAT(t.task\_name) AS List\_of\_Task*

*FROM*

*USER u*

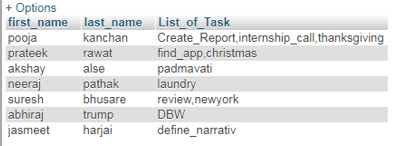
*JOIN task t ON*

*u.user\_id = t.user\_id*

*GROUP BY*

*u.user\_id*

**RESULT:**



**8.Select list name which has been assigned to minimum number of tasks than any other list.**

**(Helps to Know what kind of lists have been assigned with minimum no of tasks)**

*SELECT*

*l.list\_name, COUNT(\*) AS No\_of\_Task*

*FROM LIST*

*l*

*JOIN task t ON*

*t.list\_id = l.list\_id*

*GROUP BY*

*l.list\_id*

*HAVING*

*COUNT(\*) <= ALL(*

*SELECT*

*COUNT(\*)*

*FROM*

*task t*

*GROUP BY*

*t.list\_id*

*)*

**RESULT:**

