Complete a team project, which demonstrates your ability to work in a team to design and implement a functional system with a database, based on what you have learned in the course. Guidelines for this are: Work in self-assembled teams of two to three students.

|  |  |  |
| --- | --- | --- |
| **#** | **Task** | **Status** |
| 1 | Document with team name, group members, and which project you will work on. If the project is your own idea, a requirements document must be submitted and approved. | done |
| 2 | Data analysis of the facts listing entities, attributes, and relationships in the data model. | Leonard |
| 3 | Conceptual data model diagram. | Leonard |
| 4 | Logical data model diagram. | Leonard |
| 5 | Identification of your external data model and data logic. (make Views – skiers, rental shop, ticket office) | sean |
| 6 | Basic layout of all application screens. (on a bar napkin if need be ) | Sean |
| 7 | Diagram of each screen used in the application. | Sean |
| 8 | SQL up/down script to implement the internal model with initial data. | Blake / Sean |
| 9 | SQL up/down Script to load/migrate in existing data. | Blake / Sean |
| 10 | SQL up/down script of data logic for the external data model. | Sean |
| 11 | Implementation of the application itself. | Christian |
| 12 | A team log recording individual and group contributions to the project, including when and by whom. | Blake / Sean |
| 13 | A slide deck of your presentation. | Christian |
| 14 | A video recording of your team presentation. | Everyone |
| 15 | A video reflection of what you learned from the experience, what you would do better if you had the time, etc. | Everyone |

|  |  |
| --- | --- |
|  |  |
| 1 | * Team Megatron!! * Blake Tindol, Sean Deery, Christian Dobish, Leonard Lasek * We will model a ski resort and track rentals and lift tickets |
| 2 |  |
| 3 | Conceptual model: |
| 4 | Logical model: |
| 5 | External data model and data logic:  Skier   * Buy a lift ticket for a specific date. Lift tickets can be half-day, full-day, week, or year. One skier can buy multiple tickets (e.g., parent buys tickets for kids). If the skier has already purchased a ticket for the same day, they will receive a warning that they already have tickets before they can buy more. * Buy a rental for a ticket. A rental includes all the equipment (skis, boots, poles, helmet, etc.) for 1 skier.   Main Office Attendant   * Sell a ticket and a rental to a skier. If the skier has already purchased a ticket for that day, the attendant will let them know how many tickets they have already bought before they purchase more. * Distribute physical tickets, which makes the ticket valid for the lift gates.   Rental Shop Attendant   * Distribute a rental and mark as taken out. * Collect rental equipment and mark as returned.   Lift Gates   * Check if a ticket is valid before opening gate |
| 6 | Basic layout of all application screens.  Skier Application Screens:   * Sign in/Register * View tickets and rental purchases * Purchase ticket * Add a rental to a ticket   Main Office Attendant Application Screens:   * Attendant sign in * Print ticket * Sell Ticket * Sell Rental   Rental Shop Attendant Application Screens:   * Attendant Sign in * View returns dashboard (remaining returns for the day) * Distribute/Collect |
| 7 | Diagram of each screen used in the application. |
| 8 | SQL up/down script to implement the internal model with initial data.  if not exists(select \* from sys.databases where name='skierdb')  create database skierdb  go    use skierdb  GO    -- DOWN  drop trigger if exists t\_ticket\_datetimes  if exists(select \* from INFORMATION\_SCHEMA.TABLE\_CONSTRAINTS  where CONSTRAINT\_NAME='fk\_ticket\_skier\_id' )  alter table tickets drop constraint fk\_ticket\_skier\_id  if exists(select \* from INFORMATION\_SCHEMA.TABLE\_CONSTRAINTS  where CONSTRAINT\_NAME='fk\_ticket\_ticket\_type\_id' )  alter table tickets drop constraint fk\_ticket\_ticket\_type\_id  if exists(select \* from INFORMATION\_SCHEMA.TABLE\_CONSTRAINTS  where CONSTRAINT\_NAME='fk\_rental\_skier\_id' )  alter table rentals drop constraint fk\_rental\_skier\_id  drop table if exists skiers  drop table if exists ticket\_types  drop table if exists tickets  drop table if exists rentals  go    -- UP Metadata  create table skiers (  skier\_id int identity not null  , skier\_firstname varchar(50) not null  , skier\_lastname varchar(50) not null  , skier\_email varchar(100) not null  , skier\_date\_of\_birth date null  , constraint pk\_skier\_id primary key(skier\_id)  , constraint u\_skier\_email unique(skier\_email)  , constraint ch\_skier\_date\_of\_birth\_gt\_1900 check(skier\_date\_of\_birth > '1900-01-01')  )    create table ticket\_types (  ticket\_type\_id int identity not null  , ticket\_type\_name varchar(50) not null  , ticket\_price money not null  , constraint pk\_ticket\_type\_id primary key(ticket\_type\_id)  )    create table tickets (  ticket\_id int identity not null  , ticket\_skier\_id int not null  , ticket\_ticket\_type\_id int not null  , ticket\_datetime\_purchased datetime not NULL default getdate()  , ticket\_datetime\_begin datetime not NULL  , ticket\_datetime\_end datetime not NULL  , constraint pk\_ticket\_id primary key(ticket\_id)  , constraint fk\_ticket\_skier\_id foreign key(ticket\_skier\_id) references skiers(skier\_id)  , constraint fk\_ticket\_ticket\_type\_id foreign key(ticket\_ticket\_type\_id) references ticket\_types(ticket\_type\_id)  )    go  create trigger t\_ticket\_datetimes  on tickets instead of insert as  BEGIN  declare @begin\_date datetime = (select ticket\_datetime\_begin from inserted)  declare @ticket\_type varchar(50) = (select ticket\_type\_name from inserted join ticket\_types on ticket\_ticket\_type\_id=ticket\_type\_id)  declare @begin\_datetime\_est datetime =  case when @ticket\_type = 'PM' then concat(cast(@begin\_date as date), ' 12:30:00')  else concat(cast(@begin\_date as date), ' 08:30:00')  end  declare @begin\_datetime\_utc datetime = dateadd(hour, 5, @begin\_datetime\_est)  declare @end\_datetime\_est datetime =  case  when @ticket\_type = 'AM' then concat(cast(@begin\_date as date), ' 12:00:00')  when @ticket\_type = 'PM' or @ticket\_type = 'One Day' then concat(cast(@begin\_date as date), ' 16:00:00')  when @ticket\_type = 'Two Day' then concat(dateadd(day, 1, cast(@begin\_date as date)), ' 16:00:00')  when @ticket\_type = 'Three Day' then concat(dateadd(day, 2, cast(@begin\_date as date)), ' 16:00:00')  when @ticket\_type = 'Season' and month(@begin\_date) > 4 then concat(year(@begin\_date)+1, '-04-10 16:00:00')  when @ticket\_type = 'Season' and month(@begin\_date) < 4 then concat(year(@begin\_date), '-04-10 16:00:00')  end  declare @end\_datetime\_utc datetime = dateadd(hour, 5, @end\_datetime\_est)  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_purchased, ticket\_datetime\_begin, ticket\_datetime\_end)  values (  (select ticket\_skier\_id from inserted),  (select ticket\_ticket\_type\_id from inserted),  (select ticket\_datetime\_purchased from inserted),  @begin\_datetime\_utc,  @end\_datetime\_utc  )  END  go    create table rentals (  rental\_id int identity not null  , rental\_skier\_id int not NULL  , rental\_datetime\_purchased datetime not null default getdate()  , rental\_datetime\_taken\_out datetime NULL  , rental\_datetime\_returned datetime null  , constraint pk\_rental\_id primary key(rental\_id)  , constraint fk\_rental\_skier\_id foreign key(rental\_skier\_id) references skiers(skier\_id)  )    go  -- Insert 50 skiers  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Johannes', 'Botha', 'jbotha0@lycos.com', '8/17/1987');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Tami', 'Youles', 'tyoules1@nymag.com', '4/17/1927');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Noami', 'Scotts', 'nscotts2@thetimes.co.uk', '7/18/1935');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Godfree', 'Rumens', 'grumens3@google.com.au', '9/29/1998');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Sharai', 'Tailour', 'stailour4@acquirethisname.com', '4/22/1960');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Blair', 'Oddey', 'boddey5@mit.edu', '8/8/2010');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Yorker', 'Hengoed', 'yhengoed6@naver.com', '5/27/1951');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Douglas', 'Laurentin', 'dlaurentin7@ezinearticles.com', '2/8/1949');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Cece', 'Courtes', 'ccourtes8@meetup.com', '6/7/1929');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Cody', 'Kobsch', 'ckobsch9@narod.ru', '7/27/2004');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Bernardine', 'Oakton', 'boaktona@miibeian.gov.cn', '12/27/1955');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Simon', 'Chedgey', 'schedgeyb@sakura.ne.jp', '12/26/2010');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Tomlin', 'Prestige', 'tprestigec@merriam-webster.com', '1/26/1963');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Mercy', 'Crowter', 'mcrowterd@spotify.com', '12/27/1981');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Rochell', 'Boyton', 'rboytone@acquirethisname.com', '5/11/1952');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Brena', 'Oakland', 'boaklandf@tumblr.com', '11/5/2008');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Elroy', 'Salmons', 'esalmonsg@huffingtonpost.com', '11/7/1990');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Geralda', 'Lampel', 'glampelh@unesco.org', '9/5/1956');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Shelbi', 'Grinter', 'sgrinteri@kickstarter.com', '9/30/2004');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Sonja', 'Neward', 'snewardj@aboutads.info', '6/3/2009');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Moyna', 'Ryal', 'mryalk@economist.com', '9/24/1962');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Max', 'Troake', 'mtroakel@biblegateway.com', '9/14/1946');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Doretta', 'Shivell', 'dshivellm@bloglovin.com', '8/24/1951');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Minny', 'Coltan', 'mcoltann@eepurl.com', '12/21/1992');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Bab', 'Rivett', 'brivetto@google.com.hk', '9/6/1939');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Nicky', 'Allender', 'nallenderp@phoca.cz', '10/31/1967');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Latia', 'McDaid', 'lmcdaidq@pinterest.com', '7/15/2018');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Elva', 'Lannen', 'elannenr@foxnews.com', '12/10/1942');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Bunnie', 'Covelle', 'bcovelles@usgs.gov', '9/1/2008');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Granthem', 'Affron', 'gaffront@unblog.fr', '12/16/1960');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Dinny', 'Nowaczyk', 'dnowaczyku@java.com', '12/31/1982');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Ambrosio', 'Pitcaithly', 'apitcaithlyv@nba.com', '4/25/1969');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Yetta', 'Reasce', 'yreascew@e-recht24.de', '4/7/1992');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Fred', 'Pringle', 'fpringlex@tamu.edu', '5/18/1980');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Belia', 'Gostage', 'bgostagey@webmd.com', '3/21/1948');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Desiri', 'Davern', 'ddavernz@com.com', '9/13/2020');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Patsy', 'Stobie', 'pstobie10@fotki.com', '7/21/1995');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Retha', 'Thaxton', 'rthaxton11@icq.com', '4/3/2022');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Rebbecca', 'Wooles', 'rwooles12@newyorker.com', '8/17/2001');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Jsandye', 'Cham', 'jcham13@hc360.com', '9/3/1948');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Thom', 'Carriage', 'tcarriage14@networksolutions.com', '11/17/1944');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Hoebart', 'Bloan', 'hbloan15@cnbc.com', '5/16/1959');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Zedekiah', 'Iwanczyk', 'ziwanczyk16@canalblog.com', '6/21/1954');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Gustave', 'Redshaw', 'gredshaw17@unblog.fr', '5/31/1938');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Fidel', 'McMorran', 'fmcmorran18@unc.edu', '8/25/1943');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Teodoor', 'Budge', 'tbudge19@yellowpages.com', '5/7/1984');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Morgana', 'Dysart', 'mdysart1a@kickstarter.com', '6/15/1966');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Dulcia', 'Pittwood', 'dpittwood1b@economist.com', '12/27/1970');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Mozes', 'Patron', 'mpatron1c@bing.com', '7/13/1997');  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth) values ('Olivette', 'Berka', 'oberka1d@forbes.com', '12/18/1962');    -- Insert Ticket Types  insert into ticket\_types (ticket\_type\_name, ticket\_price) values  ('AM', 50)  , ('PM', 50)  , ('One Day', 70)  , ('Two Day', 130)  , ('Three Day', 190)  , ('Season', 500)    -- Insert 70 tickets  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (50, 2, '2022-04-07');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (43, 4, '2022-05-17');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (1, 1, '2022-08-21');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (38, 6, '2022-08-25');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (48, 1, '2022-09-03');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (10, 3, '2023-01-23');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (8, 5, '2022-12-01');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (1, 1, '2022-07-02');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (37, 2, '2022-04-14');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (47, 1, '2022-11-04');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (21, 5, '2022-09-03');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (30, 5, '2022-08-30');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (27, 1, '2023-02-26');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (16, 2, '2022-10-13');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (38, 5, '2023-02-19');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (42, 4, '2023-03-23');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (17, 3, '2022-12-26');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (41, 2, '2022-08-18');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (47, 4, '2022-11-08');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (5, 2, '2022-05-21');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (20, 5, '2023-03-04');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (38, 5, '2022-05-27');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (16, 2, '2022-10-04');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (24, 2, '2022-08-30');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (47, 5, '2022-12-04');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (47, 2, '2022-10-18');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (8, 3, '2022-10-29');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (31, 2, '2022-07-07');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (19, 3, '2022-10-05');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (47, 6, '2022-06-29');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (34, 6, '2023-02-23');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (46, 2, '2023-01-16');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (7, 6, '2022-12-05');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (16, 2, '2022-04-11');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (22, 3, '2022-03-23');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (19, 2, '2022-10-02');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (4, 1, '2022-04-13');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (35, 3, '2023-03-09');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (47, 2, '2022-08-18');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (28, 4, '2023-01-15');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (37, 5, '2022-06-02');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (18, 2, '2022-11-03');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (3, 6, '2022-03-18');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (5, 6, '2022-12-05');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (45, 3, '2022-06-25');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (3, 4, '2022-06-29');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (18, 3, '2022-10-02');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (15, 6, '2023-03-22');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (23, 3, '2022-07-26');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (14, 2, '2022-03-18');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (28, 4, '2022-11-07');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (33, 3, '2022-10-20');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (18, 2, '2022-05-18');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (12, 1, '2022-11-06');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (24, 2, '2023-01-29');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (33, 6, '2022-07-23');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (1, 2, '2022-04-02');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (32, 2, '2022-11-09');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (6, 2, '2023-02-22');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (19, 6, '2022-05-23');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (48, 1, '2022-11-05');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (38, 2, '2022-11-04');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (3, 2, '2022-12-16');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (30, 4, '2022-12-31');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (37, 6, '2022-08-01');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (18, 2, '2022-12-31');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (18, 4, '2022-12-15');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (8, 3, '2022-10-17');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (35, 1, '2022-10-14');  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin) values (42, 2, '2022-08-03');    -- Insert 60 Rentals  insert into rentals (rental\_skier\_id) values (24);  insert into rentals (rental\_skier\_id) values (31);  insert into rentals (rental\_skier\_id) values (4);  insert into rentals (rental\_skier\_id) values (24);  insert into rentals (rental\_skier\_id) values (16);  insert into rentals (rental\_skier\_id) values (2);  insert into rentals (rental\_skier\_id) values (11);  insert into rentals (rental\_skier\_id) values (28);  insert into rentals (rental\_skier\_id) values (35);  insert into rentals (rental\_skier\_id) values (40);  insert into rentals (rental\_skier\_id) values (22);  insert into rentals (rental\_skier\_id) values (25);  insert into rentals (rental\_skier\_id) values (7);  insert into rentals (rental\_skier\_id) values (30);  insert into rentals (rental\_skier\_id) values (33);  insert into rentals (rental\_skier\_id) values (37);  insert into rentals (rental\_skier\_id) values (42);  insert into rentals (rental\_skier\_id) values (23);  insert into rentals (rental\_skier\_id) values (24);  insert into rentals (rental\_skier\_id) values (4);  insert into rentals (rental\_skier\_id) values (20);  insert into rentals (rental\_skier\_id) values (10);  insert into rentals (rental\_skier\_id) values (20);  insert into rentals (rental\_skier\_id) values (39);  insert into rentals (rental\_skier\_id) values (2);  insert into rentals (rental\_skier\_id) values (7);  insert into rentals (rental\_skier\_id) values (17);  insert into rentals (rental\_skier\_id) values (27);  insert into rentals (rental\_skier\_id) values (1);  insert into rentals (rental\_skier\_id) values (1);  insert into rentals (rental\_skier\_id) values (1);  insert into rentals (rental\_skier\_id) values (37);  insert into rentals (rental\_skier\_id) values (22);  insert into rentals (rental\_skier\_id) values (17);  insert into rentals (rental\_skier\_id) values (5);  insert into rentals (rental\_skier\_id) values (35);  insert into rentals (rental\_skier\_id) values (28);  insert into rentals (rental\_skier\_id) values (20);  insert into rentals (rental\_skier\_id) values (37);  insert into rentals (rental\_skier\_id) values (12);  insert into rentals (rental\_skier\_id) values (24);  insert into rentals (rental\_skier\_id) values (38);  insert into rentals (rental\_skier\_id) values (11);  insert into rentals (rental\_skier\_id) values (45);  insert into rentals (rental\_skier\_id) values (35);  insert into rentals (rental\_skier\_id) values (28);  insert into rentals (rental\_skier\_id) values (37);  insert into rentals (rental\_skier\_id) values (39);  insert into rentals (rental\_skier\_id) values (24);  insert into rentals (rental\_skier\_id) values (19);  insert into rentals (rental\_skier\_id) values (33);  insert into rentals (rental\_skier\_id) values (49);  insert into rentals (rental\_skier\_id) values (4);  insert into rentals (rental\_skier\_id) values (35);  insert into rentals (rental\_skier\_id) values (20);  insert into rentals (rental\_skier\_id) values (44);  insert into rentals (rental\_skier\_id) values (22);  insert into rentals (rental\_skier\_id) values (27);  insert into rentals (rental\_skier\_id) values (31);  insert into rentals (rental\_skier\_id) values (11);    go  -- Verify    select TOP 5 \* from skiers  select \* from ticket\_types  select TOP 5 \* from tickets  select TOP 5 \* from rentals |
| 9 | SQL up/down Script to load/migrate in existing data.  # python data creation script with sql bulk insert statement.  #! pip install faker    from faker import Faker  import random  import datetime  import pyodbc  import pandas as pd  import json  import sqlalchemy  from sqlalchemy import create\_engine    fake = Faker()    # Generate data for skiers table  skiers\_data = []  for i in range(1, 101):  skier\_id = i  skier\_email = fake.email()  skier\_firstname = fake.first\_name()  skier\_lastname = fake.last\_name()  skier\_dob = fake.date\_of\_birth(minimum\_age=18, maximum\_age=70)  skiers\_data.append((skier\_id, skier\_email, skier\_firstname, skier\_lastname, skier\_dob))    # Generate data for ticket\_types table  ticket\_types\_data = []  for i in range(1, 6):  ticket\_type\_id = i  ticket\_type = fake.word()  ticket\_types\_data.append((ticket\_type\_id, ticket\_type))    # Generate data for rental\_types table  rental\_types\_data = []  for i in range(1, 6):  rental\_type\_id = i  rental\_type = fake.word()  rental\_types\_data.append((rental\_type\_id, rental\_type))    # Generate data for tickets table  tickets\_data = []  for i in range(1, 101):  ticket\_id = i  ticket\_skier\_id = random.randint(1, 100)  ticket\_type\_id = random.randint(1, 5)  ticket\_date\_purchase = fake.date\_between(start\_date='-1y', end\_date='today')  ticket\_datetime\_begin = fake.date\_between(start\_date='-1y', end\_date='today')  ticket\_datetime\_end = ticket\_datetime\_begin + datetime.timedelta(hours=4)  ticket\_price = round(random.uniform(50, 200), 2)  ticket\_is\_valid = fake.boolean()  tickets\_data.append((ticket\_id, ticket\_skier\_id, ticket\_type\_id, ticket\_date\_purchase, ticket\_datetime\_begin, ticket\_datetime\_end, ticket\_price, ticket\_is\_valid))    # Generate data for rentals table  rentals\_data = []  for i in range(1, 101):  rental\_id = i  rental\_skier\_id = random.randint(1, 100)  rental\_datetime\_begin = fake.date\_between(start\_date='-1y', end\_date='today')  rental\_datetime\_end = rental\_datetime\_begin + datetime.timedelta(hours=4)  rental\_return\_status = fake.word()  rentals\_data.append((rental\_id, rental\_skier\_id, rental\_datetime\_begin, rental\_datetime\_end, rental\_return\_status))    # Print the generated data  print("skiers\_data:", skiers\_data)  print("ticket\_types\_data:", ticket\_types\_data)  print("rental\_types\_data:", rental\_types\_data)  print("tickets\_data:", tickets\_data)  print("rentals\_data:", rentals\_data)  # Convert data to dataframes  skiers\_df = pd.DataFrame(skiers\_data, columns=['skier\_id', 'skier\_email', 'skier\_firstname', 'skier\_lastname', 'skier\_dob'])  ticket\_types\_df = pd.DataFrame(ticket\_types\_data, columns=['ticket\_type\_id', 'ticket\_type'])  rental\_types\_df = pd.DataFrame(rental\_types\_data, columns=['rental\_type\_id', 'rental\_type'])  tickets\_df = pd.DataFrame(tickets\_data, columns=['ticket\_id', 'ticket\_skier\_id', 'ticket\_type\_id', 'ticket\_date\_purchase', 'ticket\_datetime\_begin', 'ticket\_datetime\_end', 'ticket\_price', 'ticket\_is\_valid'])  rentals\_df = pd.DataFrame(rentals\_data, columns=['rental\_id', 'rental\_skier\_id', 'rental\_datetime\_begin', 'rental\_datetime\_end', 'rental\_return\_status'])  # Save dataframes as CSV files  skiers\_df.to\_csv('skiers.csv', index=False)  ticket\_types\_df.to\_csv('ticket\_types.csv', index=False)  rental\_types\_df.to\_csv('rental\_types.csv', index=False)  tickets\_df.to\_csv('tickets.csv', index=False)  rentals\_df.to\_csv('rentals.csv', index=False)  # Specifying the ODBC driver, server name, database, etc. directly  conn = pyodbc.connect('Driver={SQL Server};Server=tcp:elevateembed.database.windows.net,1433;Database=elevate-embedded;Uid=btindol;Pwd=10Canada2022;Encrypt=yes;TrustServerCertificate=no;Connection Timeout=30;')  cursor = conn.cursor()  # Create a SQLAlchemy engine object from the PyODBC connection object  engine = create\_engine("mssql+pyodbc://", creator=lambda: conn)  ticket\_types\_df.to\_sql('ticket\_types', con=engine, if\_exists='replace', index=False, dtype={  'ticket\_type\_id': sqlalchemy.types.Integer(),  'ticket\_type': sqlalchemy.types.String()  })  rental\_types\_df.to\_sql('rental\_types', con=engine, if\_exists='replace', index=False, dtype={  'rental\_type\_id': sqlalchemy.types.Integer(),  'rental\_type': sqlalchemy.types.String()  })  skiers\_df.to\_sql('skiers', con=engine, if\_exists='replace', index=False, dtype={  'skier\_id': sqlalchemy.types.Integer(),  'skier\_email': sqlalchemy.types.String(),  'skier\_firstname': sqlalchemy.types.String(),  'skier\_lastname': sqlalchemy.types.String(),  'skier\_dob': sqlalchemy.types.Date()  })  tickets\_df.to\_sql('tickets', con=engine, if\_exists='replace', index=False, dtype={  'ticket\_id': sqlalchemy.types.Integer(),  'ticket\_type\_id': sqlalchemy.types.Integer(),  'ticket\_start': sqlalchemy.types.DateTime(),  'ticket\_end': sqlalchemy.types.DateTime(),  'ticket\_skier\_id': sqlalchemy.types.Integer()  })  rentals\_df.to\_sql('rentals', con=engine, if\_exists='replace', index=False, dtype={  'rental\_id': sqlalchemy.types.Integer(),  'rental\_type\_id': sqlalchemy.types.Integer(),  'rental\_start': sqlalchemy.types.DateTime(),  'rental\_end': sqlalchemy.types.DateTime(),  'rental\_skier\_id': sqlalchemy.types.Integer()  })  ##################################################  # Option 2 or 3  # You can do this or sql import wizard in azure studio or explicitly use sql to bulk insert.    ##################################################  # BULK INSERT skiers  # FROM 'C:/temp/skiers.csv'  # WITH  # (  # FIELDTERMINATOR = ',',  # ROWTERMINATOR = '\n',  # FIRSTROW = 2  # );    # BULK INSERT ticket\_types  # FROM 'C:/temp/ticket\_types.csv'  # WITH  # (  # FIELDTERMINATOR = ',',  # ROWTERMINATOR = '\n',  # TABLOCK  # );    # BULK INSERT rental\_types  # FROM 'C:/temp/rental\_types.csv'  # WITH  # (  # FIELDTERMINATOR = ',',  # ROWTERMINATOR = '\n',  # TABLOCK  # );    # BULK INSERT tickets  # FROM 'C:/temp/tickets.csv'  # WITH  # (  # FIELDTERMINATOR = ',',  # ROWTERMINATOR = '\n',  # TABLOCK  # );    # BULK INSERT rentals  # FROM 'C:/temp/rentals.csv'  # WITH  # (  # FIELDTERMINATOR = ',',  # ROWTERMINATOR = '\n',  # TABLOCK  # ); |
| 10 | SQL up/down script of data logic for the external data model.  use ski\_resort  GO    -- DOWN  drop view if exists v\_manager  drop view if exists v\_attendant  drop procedure if exists p\_sign\_up  drop procedure if exists p\_sell\_ticket  drop procedure if exists p\_sell\_rental  drop procedure if exists p\_activate\_rental  drop procedure if exists p\_deactivate\_rental  drop view if exists v\_lift  GO    -- UP Metadata    -- RESORT MANAGER    -- View to see overall activity. For the resort manager app.  create view v\_manager AS  select s.\*, t.\*, tt.\*, r.\* from skiers s  join tickets t on s.skier\_id = t.ticket\_skier\_id  join ticket\_types tt on t.ticket\_ticket\_type\_id = tt.ticket\_type\_id  join rentals r on s.skier\_id = r.rental\_skier\_id  go    -- MAIN OFFICE    -- View to see current and upcoming tickets and rentals. For the skier app and main office app.  create view v\_attendant AS  select s.\*, t.\*, tt.\*, r.\* from skiers s  join tickets t on s.skier\_id = t.ticket\_skier\_id  join ticket\_types tt on t.ticket\_ticket\_type\_id = tt.ticket\_type\_id  join rentals r on s.skier\_id = r.rental\_skier\_id  where ticket\_datetime\_begin >= getdate()  go    -- Stored procedure to create a skier account, for skier app and main office app.  create procedure p\_sign\_up (  @skier\_firstname varchar(50)  , @skier\_lastname varchar(50)  , @skier\_email varchar(100)  , @skier\_date\_of\_birth date  ) as BEGIN  insert into skiers (skier\_firstname, skier\_lastname, skier\_email, skier\_date\_of\_birth)  values (@skier\_firstname, @skier\_lastname, @skier\_email, @skier\_date\_of\_birth)  end  go    -- Sell a lift ticket procedure  create procedure p\_sell\_ticket (  @ticket\_skier\_id INT  , @ticket\_ticket\_type\_id INT  , @ticket\_datetime\_begin datetime  ) as BEGIN  insert into tickets (ticket\_skier\_id, ticket\_ticket\_type\_id, ticket\_datetime\_begin)  values (@ticket\_skier\_id, @ticket\_ticket\_type\_id, @ticket\_datetime\_begin)  end  go    -- Sell a rental stored procedure  create procedure p\_sell\_rental (  @rental\_skier\_id INT  ) as BEGIN  insert into rentals (rental\_skier\_id) values (@rental\_skier\_id)  END  go    -- RENTAL SHOP    -- Procedure to distribute and activate rental  create procedure p\_activate\_rental (  @ticket\_id int  ) as BEGIN  update rentals  set rental\_datetime\_taken\_out=getdate()  from rentals r  join skiers s on r.rental\_skier\_id=s.skier\_id  join tickets t on t.ticket\_skier\_id=s.skier\_id  where ticket\_id=@ticket\_id  END  go    -- Procedure to collect and deactivate rental  create procedure p\_deactivate\_rental (  @ticket\_id int  ) as BEGIN  update rentals  set rental\_datetime\_returned=getdate()  from rentals r  join skiers s on r.rental\_skier\_id=s.skier\_id  join tickets t on t.ticket\_skier\_id=s.skier\_id  where ticket\_id=@ticket\_id  END  go    -- LIFT GATES    -- View to validate ticket and open gate  create view v\_lift AS  select \* from tickets where ticket\_datetime\_end is null and ticket\_datetime\_begin >= getdate()  go |
| 11 | Implementation of the application itself.  /\*  This implementation is for when the Griffins visit the ski resort  \*/    use skierdb  GO    /\*  Peter Griffin decides to take a family out for a ski day to our resort, which he has never been to before.  He calls into the main office and purchases tickets and rentals from the main office attendant.  \*/    -- First the main attendant sets up a skier account for Peter  delete from skiers where skier\_email='peter@familyguy.com'  declare @skier\_id int  exec @skier\_id = p\_sign\_up @skier\_firstname='Peter', @skier\_lastname='Griffin', @skier\_email='peter@familyguy.com', @skier\_date\_of\_birth='1966-09-22'    select \* from skiers where skier\_email='peter@familyguy.com'    -- Next, the main attendant sells 6 One Day tickets and 6 rentals since none of the Griffins own skis.  declare @ticket\_id1 int  declare @ticket\_id2 int  declare @ticket\_id3 int  declare @ticket\_id4 int  declare @ticket\_id5 int  declare @ticket\_id6 int  exec @ticket\_id1 = p\_sell\_ticket @ticket\_skier\_id=@skier\_id, @ticket\_ticket\_type\_id=3, @ticket\_datetime\_begin='2023-03-27'  exec @ticket\_id2 = p\_sell\_ticket @ticket\_skier\_id=@skier\_id, @ticket\_ticket\_type\_id=3, @ticket\_datetime\_begin='2023-03-27'  exec @ticket\_id3 = p\_sell\_ticket @ticket\_skier\_id=@skier\_id, @ticket\_ticket\_type\_id=3, @ticket\_datetime\_begin='2023-03-27'  exec @ticket\_id4 = p\_sell\_ticket @ticket\_skier\_id=@skier\_id, @ticket\_ticket\_type\_id=3, @ticket\_datetime\_begin='2023-03-27'  exec @ticket\_id5 = p\_sell\_ticket @ticket\_skier\_id=@skier\_id, @ticket\_ticket\_type\_id=3, @ticket\_datetime\_begin='2023-03-27'  exec @ticket\_id6 = p\_sell\_ticket @ticket\_skier\_id=@skier\_id, @ticket\_ticket\_type\_id=3, @ticket\_datetime\_begin='2023-03-27'  exec p\_sell\_rental @rental\_skier\_id=@skier\_id  exec p\_sell\_rental @rental\_skier\_id=@skier\_id  exec p\_sell\_rental @rental\_skier\_id=@skier\_id  exec p\_sell\_rental @rental\_skier\_id=@skier\_id  exec p\_sell\_rental @rental\_skier\_id=@skier\_id  exec p\_sell\_rental @rental\_skier\_id=@skier\_id    -- The Griffins drive to the resort and they show up to the main office to collect their lift tickets.  -- The main office attendant is able to look up the skier's tickets and print out the correct ones by the begin/end dates.  select \* from v\_attendant\_tickets where skier\_email='peter@familyguy.com'    -- The Griffins go from the main office over to the rental shop and scan their tickets to receive and activate their rentals  select \* from rentals where rental\_skier\_id=@skier\_id    exec p\_activate\_rental @ticket\_id=@ticket\_id1  exec p\_activate\_rental @ticket\_id=@ticket\_id2  exec p\_activate\_rental @ticket\_id=@ticket\_id3  exec p\_activate\_rental @ticket\_id=@ticket\_id4  exec p\_activate\_rental @ticket\_id=@ticket\_id5  exec p\_activate\_rental @ticket\_id=@ticket\_id6    select \* from rentals where rental\_skier\_id=@skier\_id    -- Stewie is evil so he tries to take a second rental so he can sell it on the black market.  -- The rental shop attendant sees that the rental is not valid and directs Stewie to the main office to purchase a rental.  -- exec p\_activate\_rental @ticket\_id=@ticket\_id6    -- The Griffins go up the lift  select \* from v\_lift where ticket\_id=@ticket\_id1  select \* from v\_lift where ticket\_id=@ticket\_id2  select \* from v\_lift where ticket\_id=@ticket\_id3  select \* from v\_lift where ticket\_id=@ticket\_id4  select \* from v\_lift where ticket\_id=@ticket\_id5  select \* from v\_lift where ticket\_id=@ticket\_id6    -- The Griffins return their rental equipment at the rental shop  -- Stewie is still evil so he does not return his rental equipment  select \* from rentals where rental\_skier\_id=@skier\_id    exec p\_deactivate\_rental @ticket\_id=@ticket\_id1  exec p\_deactivate\_rental @ticket\_id=@ticket\_id2  exec p\_deactivate\_rental @ticket\_id=@ticket\_id3  exec p\_deactivate\_rental @ticket\_id=@ticket\_id4  exec p\_deactivate\_rental @ticket\_id=@ticket\_id5    select \* from rentals where rental\_skier\_id=@skier\_id    -- The main office auto attendent checks if all the rentals were returned  select \* from v\_attendant\_rentals where rental\_datetime\_taken\_out is not null and rental\_datetime\_returned is null |
|  |  |
|  |  |
|  |  |
|  |  |

The rubric is as follows:

1. Quality of artifacts (slides, code, video, documents, etc.)

2. Completeness of project—does it work?

3. Purpose of project—does it do something of value?

4. What was learned from the experience?

5. Does the project demonstrate what the team has learned pertinent to the course?