```sql
select 'hello, sql' as greeting


--creating the user table
CREATE TABLE topshot_user(
    --create cols
    topshot_user_id int identity
    , topshot_username varchar(50) NOT NULL
    , user_join_date datetime NOT NULL
    , showcases_count int
    , moments_count int
    , user_team varchar(50)
    --place contraints
    , CONSTRAINT PK_topshot_user PRIMARY KEY (topshot_user_id)
    , CONSTRAINT U1_topshot_user UNIQUE(topshot_username)
)
-- End Creating the User table

--Creating Moments table
CREATE TABLE moments(
    --create cols
    moment_unique_id int identity
    , serial_number int NOT NULL
    , player_name varchar(200) NOT NULL
    , game_date datetime NOT NULL
    , pack_name varchar(100) NOT NULL
    , highlight_type varchar(100) NOT NULL
    , moment_type varchar(100) NOT NULL
    , circulation_count int NOT NULL
    , circulation_status varchar(50) NOT NULL
    , highlight_season varchar(50) NOT NULL
    -- place constraints
    , CONSTRAINT PK_moments PRIMARY KEY(moment_unique_id)
)

--Creating table with User's moments
CREATE TABLE user_moments_collection(
    --creating cols
    collection_id int identity
    , topshot_user_id int NOT NULL
    , moment_unique_id int NOT NULL
    , purchase_date datetime NOT NULL
    --place constraints
    , CONSTRAINT PK_user_moments_collection PRIMARY KEY(collection_id)
    , CONSTRAINT FK1_user_moments_collection FOREIGN KEY(moment_unique_id)
      REFERENCES moments(moment_unique_id)
    , CONSTRAINT FK2_user_moments_collection FOREIGN KEY(topshot_user_id)
      REFERENCES topshot_user(topshot_user_id)
    , CONSTRAINT U1_user_moments_collection UNIQUE(moment_unique_id)
```

```sql
)

DROP TABLE user_moments_collection

--Creating purchase order table for Moments
CREATE TABLE moments_purchase_order(
    moment_po_transaction_id int identity
    , topshot_user_id int NOT NULL
    , moment_unique_id int NOT NULL
    , moment_purchase_price int NOT NULL
    , payment_type varchar(50) NOT NULL
    --place contraints
    , CONSTRAINT PK_moments_purchase_order PRIMARY KEY (moment_po_transaction_id)
    ,  CONSTRAINT FK1_moments_purchase_order FOREIGN KEY(topshot_user_id)
      REFERENCES topshot_user(topshot_user_id)
    ,  CONSTRAINT FK2_moments_purchase_order FOREIGN KEY(moment_unique_id)
      REFERENCES moments(moment_unique_id)
)

--Creating table for Packs
CREATE TABLE packs(
    pack_id int identity
    , pack_type varchar(50) NOT NULL
    , moments_count int NOT NULL
    , drop_number int NOT NULL
    , pack_circulation_status varchar(50)
    , pack_name varchar(50)
    -- place constraints
    CONSTRAINT PK_packs PRIMARY KEY(pack_id)
)

-- Creating table for Packs Purchase Order
CREATE TABLE pack_purchase_order(
    pack_po_transaction_id int identity
    , topshot_user_id int NOT NULL
    , pack_id int NOT NULL
    , pack_purchase_price int NOT NULL
    , payment_type varchar(50) NOT NULL
    --place constraints
    , CONSTRAINT PK_pack_purchase_order PRIMARY KEY (pack_po_transaction_id)
    , CONSTRAINT FK1_pack_purchase_order FOREIGN KEY(topshot_user_id) REFERENCES
      topshot_user(topshot_user_id)
    , CONSTRAINT FK2_pack_purchase_order FOREIGN KEY(pack_id) REFERENCES packs
      (pack_id)
)

--Creating table for Flow blockchain that validates all transactions
CREATE TABLE flow_blockchain(
    ledger_transaction_id int identity
```

```
	, order_timestamp datetime NOT NULL
	, pack_id int
	, pack_po_transaction_id int
	, moment_po_transaction_id int
	, moment_unique_id int
	--place constraints
	, CONSTRAINT PK_flow_blockchain PRIMARY KEY(ledger_transaction_id)
	, CONSTRAINT FK1_flow_blockchain FOREIGN KEY(pack_id) REFERENCES packs(pack_id) 
	, CONSTRAINT FK2_flow_blockchain FOREIGN KEY(pack_po_transaction_id) REFERENCES  pack_purchase_order(pack_po_transaction_id)
	, CONSTRAINT FK3_flow_blockchain FOREIGN KEY(moment_po_transaction_id) REFERENCES moments_purchase_order(moment_po_transaction_id)
	, CONSTRAINT FK4_flow_blockchain FOREIGN KEY(moment_unique_id) REFERENCES moments(moment_unique_id)
)


-- There is a bug in our payment system that need to be resolved,
-- it's only affecting the base set packs so we have to change these from available to unavailable.

SELECT * FROM packs

-- We do not have the unavailable pack circulation status so we have to insert that into our table first

--adding unavailable status, so we know which packs are not sold out, and available for purchase once we fix the bug
UPDATE packs SET pack_circulation_status = 'unavailable' WHERE pack_circulation_status = 'available'

SELECT * FROM packs

--cool, now our ecommerce platform has marked our available base set cards to unavailable,
--that way once the bug is fixed we can switch the unavails back to avail


-- insert our users into the topshot_user table
insert into topshot_user (user_join_date, topshot_username, showcases_count, moments_count, user_team)
values
('2020-06-15 02:06:31', 'rbrownsmith0', 6, 1184, null)
,('2020-02-06 02:32:01', 'anesbeth1', 6, 54, 'Phoenix Suns')
,('2019-08-26 20:40:08', 'tcustance2', null, 1261, 'Milwaukee Bucks')
, ('2020-11-17 20:10:35', 'glamey3', 10, 903, 'San Antonio Spurs')
, ('2020-04-04 16:43:18', 'braffels4', null, 293, 'Indiana Pacers')
, ('2020-08-26 07:04:01', 'lwessing5', 9, 1030, 'Washington Wizards')
```

```sql
, ('2019-09-05 14:18:15', 'anazer6', 2, 974, null)
, ('2019-08-03 20:25:05', 'llawey7', 4, 552, 'Golden State Warriors')
, ('2019-08-26 02:46:36', 'mboswward8', 6, 1234, null)
, ('2019-08-27 09:19:55', 'hhasted9', null, 354, 'Phoenix Suns')
, ('2020-01-27 01:50:29', 'gaugustea', null, 678, 'Oklahoma City Thunder')
, ('2020-10-03 07:46:36', 'cponcefordb', null, 605, 'Portland Trail Blazers')
, ('2020-01-15 14:50:08', 'tsperac', null, 635, 'San Antonio Spurs')
, ('2020-07-16 21:52:48', 'hnandd', 10, 460, 'Phoenix Suns')
, ('2019-07-15 17:54:12', 'tmullingere', 10, 893, 'Los Angeles Lakers')
, ('2020-10-21 04:18:11', 'dhearnef', null, 1093, 'New York Knicks')
, ('2019-10-21 00:41:34', 'murlingg', null, 1006, null)
, ('2020-09-08 10:57:29', 'tskinh', 1, 212, 'Utah Jazz')
, ('2020-08-22 06:19:17', 'ldoughtyi', 3, 789, 'Atlanta Hawks')
, ('2020-09-03 16:17:18', 'cberthomierj', null, 893, 'Utah Jazz')
, ('2020-03-19 12:46:33', 'zvelak', 4, 866, 'Cleveland Cavaliers')
, ('2020-10-15 20:29:09', 'leynaldl', null, 216, 'Los Angeles Clippers')
, ('2019-10-12 19:57:04', 'pbarnwallm', null, 240, 'Dallas Mavericks')

--Let's check our data!

SELECT * FROM topshot_user

--We have users now! Lot's of them have not created showcases, some have not
  selected their favorite team, and one has not purchases any moments!

-- Insert data officially licensed nba highlights into the moments table
insert into moments  (serial_number, player_name , game_date , pack_name ,
  highlight_type , moment_type , circulation_count, circulation_status ,
  highlight_season)
values (39, 'Steven Adams', '07/29/2020', 'base set', 'layup', 'rare', 3326, 'cc',
  '2019-2020')
,(94, 'Paul Millsap', '08/20/2020', 'metallic hold le', 'block', 'common', 3326,
  'cc', '2018-2019')
,(171, 'OG Anunoby', '10/02/2020', 'metallic hold le', 'steal', 'common', 299,
  'le', '2019-2020')
,(8, 'Royce O''Neale', '08/31/2020', 'so fresh', 'steal', 'common', 299, 'cc',
  '2019-2020')
,(90, 'Joe Harris', '10/07/2020', 'so fresh', 'handles', 'rare', 299, 'cc',
  '2019-2020')
,(180, 'Nemanja Bjelica', '10/16/2020', 'metallic hold le', 'handles', 'common',
  275, 'le', '2019-2020')
,(74, 'Miles Bridges', '08/08/2020', 'premium pack', 'handles', 'common', 2343,
  'le', '2019-2020')
,(55, 'Caris LeVert', '07/28/2020', 'metallic hold le', 'handles', 'common', 3326,
  'cc', '2019-2020')
,(242, 'Jonathan Isaac', '08/09/2020', 'base set', 'dunk', 'common', 345, 'cc',
  '2019-2020')
,(6, 'Tyler Herro', '10/16/2020', 'metallic hold le', 'steal', 'common', 3134,
  'le', '2019-2020')
```

```
,(138, 'Patty Mills', '07/27/2020', 'premium pack', 'layup', 'common', 3134, 'cc', 
  '2019-2020')
,(123, 'OG Anunoby', '08/08/2020', 'lace em'' up', 'handles', 'common', 1232, 'cc', 
   '2019-2020')
,(79, 'Derrick White', '08/07/2020', 'so fresh', 'steal', 'common', 150, 'cc', 
  '2018-2019')
,(12, 'Willie Cauley-Stein', '09/28/2020', 'premium pack', 'assist', 'common', 
  3134, 'le', '2019-2020')
,(20, 'Seth Curry', '07/30/2020', 'so fresh', 'dunk', 'common', 3134, 'cc', 
  '2019-2020')
,(173, 'Doug McDermott', '10/10/2020', 'lace em'' up', 'block', 'common', 3326, 
  'cc', '2018-2019')
,(30, 'Brandon Clarke', '10/02/2020', 'premium pack', 'dunk', 'common', 299, 'le', 
  '2019-2020')
,(240, 'John Henson', '09/20/2020', 'base set', 'steal', 'common', 3134, 'le', 
  '2018-2019')
,(295, 'Bogdan Bogdanovic', '08/11/2020', 'premium pack', 'dunk', 'common', 3134, 
  'cc', '2019-2020')
,(255, 'Patty Mills', '08/21/2020', 'base set', 'steal', 'common', 2343, 'le', 
  '2019-2020')
,(238, 'Bojan Bogdanovic', '07/23/2020', 'base set', 'handles', 'rare', 150, 'cc', 
  '2019-2020')
,(114, 'Donovan Mitchell', '07/22/2020', 'lace em'' up', 'assist', 'rare', 299, 
  'cc', '2019-2020')
,(137, 'Pascal Siakam', '10/09/2020', 'metallic hold le', 'jump shot', 'common', 
  3134, 'cc', '2019-2020')
,(46, 'Jae Crowder', '08/07/2020', 'metallic hold le', 'block', 'common', 150, 
  'cc', '2019-2020')
,(36, 'Ish Smith', '07/28/2020', 'so fresh', 'dunk', 'common', 1232, 'cc', 
  '2018-2019')
,(280, 'P.J. Tucker', '10/12/2020', 'lace em'' up', 'dunk', 'rare', 345, 'le', 
  '2019-2020')
,(138, 'RJ Barrett', '10/12/2020', 'metallic hold le', 'layup', 'common', 2343, 
  'le', '2019-2020')
,(281, 'Pascal Siakam', '09/15/2020', 'metallic hold le', 'block', 'common', 3134, 
  'cc', '2019-2020')
,(249, 'Doug McDermott', '09/24/2020', 'metallic hold le', 'block', 'common', 2432, 
   'cc', '2019-2020')
,(121, 'Aron Baynes', '09/03/2020', 'so fresh', 'handles', 'common', 2343, 'cc', 
  '2019-2020')
,(79, 'Kenneth Faried', '07/29/2020', 'so fresh', 'jump shot', 'common', 1232, 
  'cc', '2019-2020')
,(138, 'Kosta Koufos', '09/27/2020', 'so fresh', 'layup', 'common', 345, 'cc', 
  '2019-2020')
,(275, 'Pascal Siakam', '09/09/2020', 'lace em'' up', 'steal', 'rare', 275, 'cc', 
  '2019-2020')
,(223, 'Dejounte Murray', '09/11/2020', 'premium pack', 'dunk', 'common', 2432, 
  'le', '2019-2020')
,(292, 'Tyler Zeller', '08/26/2020', 'base set', 'assist', 'rare', 2432, 'cc', 
```

```sql
'2019-2020')
,(65, 'Denzel Valentine', '10/08/2020', 'base set', 'assist', 'common', 345, 'cc', '2018-2019')

--Lots of rows, but that's bc we want a large inventory of highlights for users to purchase!

--Now let's insert moments to user's collections that were already purchased from July - November 2020

insert into user_moments_collection (topshot_user_id, moment_unique_id, purchase_date)
values(25, 2953, '8/8/2020')
,(18, 2007, '11/19/2020')
, (4, 2534, '10/30/2020')
,(10, 2505, '7/21/2020')
,(1, 2216, '7/13/2020')
, (1, 2783, '11/24/2020')
,(3, 2385, '11/18/2020')
,(14, 2524, '8/7/2020')
,(6, 2373, '11/29/2020')
,(8, 2155, '11/1/2020')
,(1, 2744, '8/20/2020')
,(6, 2168, '11/5/2020')
,(12, 2204, '7/25/2020')
,(10, 2419, '8/27/2020')
,(12, 2876, '7/18/2020')
,(12, 2737, '8/15/2020')
,(4, 2082, '8/8/2020')
,(3, 2332, '10/3/2020')
,(7, 2063, '8/15/2020')
,(18, 2925, '9/5/2020')
,(18, 2962, '9/14/2020')
,(5, 2681, '7/24/2020')
,(16, 2420, '7/15/2020')
,(1, 2916, '8/5/2020')
,(25, 2150, '7/23/2020')
,(18, 2794, '11/19/2020')
,(18, 2578, '7/20/2020')
,(4, 2675, '8/5/2020')
,(8, 2046, '9/28/2020')
,(14, 2278, '10/9/2020')
,(6, 2686, '10/2/2020')
,(14, 2712, '7/13/2020')
, (12, 2410, '9/29/2020')
,(17, 2395, '7/4/2020')
,(10, 2615, '11/23/2020')
,(15, 2110, '11/13/2020')
,(19, 2283, '11/14/2020')
```

```sql
,(3, 2209, '7/6/2020')
,(17, 2493, '10/27/2020')
,(11, 2132, '10/15/2020')
,(21, 2343, '8/6/2020')
,(12, 2428, '11/8/2020')
,(4, 2279, '8/25/2020')
,(23, 2223, '11/26/2020')
,(3, 2695, '11/24/2020')
,(7, 2259, '9/15/2020')
,(9, 2598, '9/25/2020')
,(6, 2701, '10/22/2020')
,(18, 2420, '11/11/2020')
,(12, 2681, '11/26/2020')
,(6, 2449, '7/12/2020')
,(10, 2885, '9/30/2020')
,(23, 2299, '11/11/2020')
,(7, 2277, '8/1/2020')
,(24, 2942, '9/30/2020')
,(19, 2353, '8/10/2020')
,(3, 2395, '9/21/2020')
,(12, 2187, '11/28/2020')
,(25, 2383, '8/31/2020')
,(2, 2064, '9/27/2020')
,(11, 2278, '10/31/2020')
,(5, 2130, '9/23/2020')
,(20, 2395, '9/2/2020')
,(23, 2980, '9/4/2020')
,(12, 2207, '8/14/2020')
, (15, 2065, '9/16/2020')
, (5, 2763, '11/2/2020')
, (21, 2085, '11/7/2020')
, (12, 2058, '9/15/2020')
, (23, 2459, '10/25/2020')
, (12, 2360, '11/24/2020')
, (3, 2519, '8/25/2020')
, (6, 2297, '7/18/2020')
, (1, 2269, '10/4/2020')
, (14, 2467, '9/10/2020')
, (13, 2274, '11/27/2020')
, (8, 2211, '10/28/2020')
, (15, 2346, '8/23/2020')
, (9, 2661, '10/2/2020')
, (25, 2846, '9/8/2020')
,(10, 2257, '10/19/2020')
, (14, 2359, '7/6/2020')
, (17, 2568, '9/18/2020')
, (11, 2667, '11/22/2020')
, (2, 2164, '7/18/2020')
, (8, 2298, '10/21/2020')
, (1, 2798, '9/6/2020')
, (13, 2688, '7/30/2020')
, (20, 2537, '9/8/2020')
, (20, 2779, '9/7/2020')
, (19, 2010, '9/23/2020')
, (5, 2707, '10/3/2020')
, (11, 2382, '10/23/2020')
, (10, 2021, '8/1/2020')
, (5, 2690, '10/14/2020')
, (17, 2984, '8/20/2020')
,(8, 2054, '8/3/2020')
,(15, 2314, '10/3/2020')
,(7, 2372, '7/3/2020')
,(21, 2254, '9/24/2020')

SELECT * FROM user_moments_collection

--Now let's insert information about packs available for purchase where users can buy their moments
insert into packs (pack_type, pack_purchase_price , moments_count , drop_number , pack_circulation_status , pack_name)
values ('common', '20', '8', 3, 'available', 'base set')
,('rare', '99', '10', 1, 'sold out', 'lace em up')
,('common', '20', '8', 1, 'available', 'base set')
,('rare', '99', '10', 3, 'sold out', 'lace em up')
,('common', '20', '8', 2, 'available', 'base set')
,('rare', '99', '10', 3, 'sold out', 'lace em up')
,('common', '20', '8', 1, 'available', 'base set')
,('common', '20', '8', 1, 'available', 'base set')
,('common', '20', '8', 1, 'available', 'base set')
,('rare', '99', '10', 1, 'sold out', 'lace em up')
,('common', '20', '8', 2, 'available', 'base set')
,('common', '20', '8', 1, 'available', 'base set')
,('common', '20', '8', 3, 'available', 'base set')
,('common', '20', '8', 1, 'available', 'base set')
,('common', '20', '8', 3, 'available', 'base set')
,('common', '20', '8', 2, 'available', 'base set')
,('common', '20', '8', 3, 'available', 'base set')
,('common', '20', '8', 3, 'available', 'base set')
,('common', '20', '8', 3, 'available', 'base set')
, ('common', '20', '8', 2, 'available', 'base set')

--Now that we have our packs for sale, let's add most recent Pack transactions
insert into pack_purchase_order (topshot_user_id, pack_id, payment_type)
values (13, 16, 'debit card')
, (1, 16, 'paypal')
, (17, 15, 'ethereum')
, (4, 10, 'ethereum')
, (5, 9, 'paypal')
, (13, 16, 'credit card')
, (17, 14, 'paypal')
, (1, 7, 'ethereum')

--Now let's add moments that were most recently purchased
insert into moments_purchase_order (topshot_user_id, moment_unique_id,
  moment_purchase_price , payment_type)
values (14, 2591, 1457, 'ethereum')
, (10, 2694, 943, 'debit card')
, (23, 2277, 1987, 'paypal')
, (23, 2865, 2145, 'debit card')
, (23, 2660, 1488, 'ethereum')
, (15, 2899, 2166, 'paypal')
, (11, 2684, 1943, 'credit card')
,(19, 2436, 1972, 'credit card')

-- finally, let's add data to the flow blockchain which validates all our
  transactions
insert into flow_blockchain  (order_timestamp , pack_id, pack_po_transaction_id,
  moment_po_transaction_id, moment_unique_id)
values ('2020-06-09 08:04:02', 9, 7, null, null)
, ('2020-03-12 07:15:13', 18, 4, 101, 2079)
, ('2020-06-26 11:01:18', 20, 2, 102, 2748)
, ('2020-07-24 01:03:28', 4, 3, 105, 2179)
, ('2019-12-05 20:20:28', 15, 1, null, null)
, ('2020-01-22 08:41:14', 2, 6, 107, 2160)
, ('2020-06-05 15:30:05', 19, 3, 104, 2497)
, ('2020-07-18 02:33:49', 2, 1, 107, 2576)
, ('2020-07-14 11:47:55', 16, 8, 104, 2149)
,('2019-12-15 22:47:53', 6, 4, 104, 2764)
, ('2020-07-09 22:57:31', 10, 2, 107, 2554)
, ('2020-07-29 11:35:46', 18, 6, 102, 2581)
, ('2020-11-21 15:19:31', 13, 8, 103, 2202)
, ('2020-08-31 04:04:07', 4, 7, 104, 2236)
, ('2020-06-12 06:11:02', 3, 7, 102, 2430)
, ('2019-12-18 16:00:18', 6, 4, 101, 2133)
, ('2020-01-23 09:10:47', 17, 7, 105, 2825)
, ('2020-08-16 14:53:31', 6, 4, 108, 2092)


--Create view of User and their moments
CREATE VIEW TopShotUserMomentsLibrary
AS
SELECT
     topshot_user.topshot_username AS UserName
     , moments.moment_unique_id as MOMENT
FROM user_moments_collection
INNER JOIN user_moments_collection ON user_moments_collection.topshot_user_id =
  topshot_user.topshot_user_id
```

```sql
GO

-- Data Question #1: From our early adopters cohort, who owms the most moments and how many?

SELECT topshot_user.user_join_date, topshot_user.topshot_username, topshot_user.user_team, topshot_user.moments_count
FROM topshot_user
WHERE moments_count > 1000
ORDER BY topshot_user.moments_count DESC;

-- Data Question #2: From the most recent pack purchase order transactions, what pack was the best selling?

SELECT packs.pack_name, packs.pack_purchase_price, topshot_user.topshot_username
FROM topshot_user
INNER JOIN (packs INNER JOIN pack_purchase_order
ON packs.pack_id = pack_purchase_order.pack_id)
ON topshot_user.topshot_user_id = pack_purchase_order.topshot_user_id;


-- Data Question #3:  From most recent moment order transactions, who spent the most money and how much?
SELECT topshot_user.topshot_username, Count(moments_purchase_order.payment_type) AS CountOfpayment_type
,Sum(moments_purchase_order.moment_purchase_price) AS SumOfmoment_purchase_price
FROM topshot_user INNER JOIN moments_purchase_order
ON topshot_user.topshot_user_id = moments_purchase_order.topshot_user_id
GROUP BY topshot_user.topshot_username
ORDER BY Count(moments_purchase_order.payment_type);

-- Data Question #4: From our most recent moment purchases, what was the average transaction per moment type by each user?
SELECT topshot_user.topshot_username,
Avg(moments_purchase_order.moment_purchase_price) AS AvgOfmoment_purchase_price, moments.moment_type
FROM moments INNER JOIN (topshot_user INNER JOIN moments_purchase_order
ON topshot_user.topshot_user_id = moments_purchase_order.topshot_user_id)
ON moments.moment_unique_id = moments_purchase_order.moment_unique_id
GROUP BY topshot_user.topshot_username, moments.moment_type
ORDER BY topshot_user.topshot_username;

--Data Question #5: What was the most popular payment type from the most recent transactions?
SELECT Count(moments_purchase_order.moment_po_transaction_id)
AS CountOfmoment_po_transaction_id, moments_purchase_order.payment_type
FROM moments_purchase_order
GROUP BY moments_purchase_order.payment_type;
```