```
 * mysql+pymysql://root:***@localhost
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
```

Out[30]:
```
[]
```

In [31]:
```
%%sql

ALTER TABLE Bill ADD FOREIGN KEY(appt_ID) references Appointments(appt_ID);
ALTER TABLE Payment ADD FOREIGN KEY(patient_ID) references Patient (patient_ID);
ALTER TABLE Payment ADD FOREIGN KEY(company_ID) references Insurance_Company (company_
ALTER TABLE Payment ADD FOREIGN KEY(bill_ID) references Bill (bill_ID);
ALTER TABLE Insurance_Company ADD FOREIGN KEY(patient_ID) references Patient (patient_
ALTER TABLE Patient ADD FOREIGN KEY(company_ID) references Insurance_Company (company_
ALTER TABLE Appointments ADD FOREIGN KEY(physician_ID) references Physician (physician
ALTER TABLE Appointments ADD FOREIGN KEY(patient_ID) references Patient (patient_ID)
```

```
 * mysql+pymysql://root:***@localhost
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
```

Out[31]:
```
[]
```

In [32]:
```
#add constraint to make sure total=insured+not insured
%sql ALTER TABLE Bill ADD CONSTRAINT check_total_amount CHECK (bill_total =amount_insu
```

```
 * mysql+pymysql://root:***@localhost
0 rows affected.
```

Out[32]:
```
[]
```

S2

Question

- *Use the classic models database that you loaded.*
- *Write a query that returns the following results:*

`(customerNumber, customerName, no_of_orders, total_revenue)`

- *where:*
  - `customerNumber` *and* `customerName` *are from* `customers`.
  - `no_of_orders` *is the number of orders the customer has placed.*

- `total_revenue` is the sum of `quantityOrdered*priceEach` for all `orderDetails` in `orders` associated with a customer.

- If a customer has not placed any orders, `no_of_orders` and `total_revenue` must be `0`.

Answer

In [33]:
```
%%sql

use classicmodels;

SELECT
    c.customerNumber,
    c.customerName,
    COALESCE(o.no_of_orders, 0) AS no_of_orders,
    COALESCE(o.total_revenue, 0) AS total_revenue
FROM
    classicmodels.customers AS c
LEFT JOIN
    (
        SELECT
            o.customerNumber,
            COUNT(DISTINCT o.orderNumber) AS no_of_orders,
            SUM(od.quantityOrdered * od.priceEach) AS total_revenue
        FROM
            classicmodels.orders AS o
        LEFT JOIN
            classicmodels.orderdetails AS od ON o.orderNumber = od.orderNumber
        GROUP BY
            o.customerNumber
    ) o ON c.customerNumber = o.customerNumber
ORDER BY
    c.customerNumber;
```

```
 * mysql+pymysql://root:***@localhost
0 rows affected.
122 rows affected.
```

Out[33]:

customerNumber	customerName	no_of_orders	total_revenue
103	Atelier graphique	3	22314.36
112	Signal Gift Stores	3	80180.98
114	Australian Collectors, Co.	5	180585.07
119	La Rochelle Gifts	4	158573.12
121	Baane Mini Imports	4	104224.79
124	Mini Gifts Distributors Ltd.	17	591827.34
125	Havel & Zbyszek Co	0	0.00
128	Blauer See Auto, Co.	4	75937.76
129	Mini Wheels Co.	3	66710.56
131	Land of Toys Inc.	4	149085.15
141	Euro+ Shopping Channel	26	820689.54
144	Volvo Model Replicas, Co	4	66694.82
145	Danish Wholesale Imports	5	129085.12
146	Saveley & Henriot, Co.	3	130305.35
148	Dragon Souveniers, Ltd.	5	156251.03
151	Muscle Machine Inc	4	177913.95
157	Diecast Classics Inc.	4	104358.69
161	Technics Stores Inc.	4	104545.22
166	Handji Gifts& Co	4	107746.75
167	Herkku Gifts	3	97562.47
168	American Souvenirs Inc	0	0.00
169	Porto Imports Co.	0	0.00
171	Daedalus Designs Imports	2	61781.70
172	La Corne D'abondance, Co.	3	86553.52
173	Cambridge Collectables Co.	2	32198.69
175	Gift Depot Inc.	3	95424.63
177	Osaka Souveniers Co.	2	62361.22
181	Vitachrome Inc.	3	72497.64
186	Toys of Finland, Co.	3	95546.46
187	AV Stores, Co.	3	148410.09
189	Clover Collections, Co.	2	49898.27
198	Auto-Moto Classics Inc.	3	21554.26
201	UK Collectables, Ltd.	3	106610.72
202	Canadian Gift Exchange Network	2	70122.19

204	Online Mini Collectables	2	55577.26
205	Toys4GrownUps.com	3	93803.30
206	Asian Shopping Network, Co	0	0.00
209	Mini Caravy	3	75859.32
211	King Kong Collectables, Co.	2	45480.79
216	Enaco Distributors	3	68520.47
219	Boards & Toys Co.	2	7918.60
223	Natürlich Autos	0	0.00
227	Heintze Collectables	2	89909.80
233	Québec Home Shopping Network	3	68977.67
237	ANG Resellers	0	0.00
239	Collectable Mini Designs Co.	2	80375.24
240	giftsbymail.co.uk	2	71783.75
242	Alpha Cognac	3	60483.36
247	Messner Shopping Network	0	0.00
249	Amica Models & Co.	2	82223.23
250	Lyon Souveniers	3	67659.19
256	Auto Associés & Cie.	2	58876.41
259	Toms Spezialitäten, Ltd	2	89223.14
260	Royal Canadian Collectables, Ltd.	2	66812.00
273	Franken Gifts, Co	0	0.00
276	Anna's Decorations, Ltd	4	137034.22
278	Rovelli Gifts	3	127529.69
282	Souveniers And Things Co.	4	133907.12
286	Marta's Replicas Co.	2	90545.37
293	BG&E Collectables	0	0.00
298	Vida Sport, Ltd	2	108777.92
299	Norway Gifts By Mail, Co.	2	69059.04
303	Schuyler Imports	0	0.00
307	Der Hund Imports	0	0.00
311	Oulu Toy Supplies, Inc.	3	95706.15
314	Petit Auto	3	70851.58
319	Mini Classics	2	78432.16
320	Mini Creations Ltd.	3	101872.52
321	Corporate Gift Ideas Co.	4	132340.78

323	Down Under Souveniers, Inc	5	154622.08
324	Stylish Desk Decors, Co.	3	80556.73
328	Tekni Collectables Inc.	3	81806.55
333	Australian Gift Network, Co	3	55190.16
334	Suominen Souveniers	3	103896.74
335	Cramer Spezialitäten, Ltd	0	0.00
339	Classic Gift Ideas, Inc	2	57939.34
344	CAF Imports	2	46751.14
347	Men 'R' US Retailers, Ltd.	2	41506.19
348	Asian Treasures, Inc.	0	0.00
350	Marseille Mini Autos	3	71547.53
353	Reims Collectables	5	126983.19
356	SAR Distributors, Co	0	0.00
357	GiftsForHim.com	3	94431.76
361	Kommission Auto	0	0.00
362	Gifts4AllAges.com	3	84340.32
363	Online Diecast Creations Co.	3	116449.29
369	Lisboa Souveniers, Inc	0	0.00
376	Precious Collectables	0	0.00
379	Collectables For Less Inc.	3	73533.65
381	Royale Belge	4	29217.18
382	Salzburg Collectables	4	137480.07
385	Cruz & Sons Co.	3	87468.30
386	L'ordine Souveniers	3	125505.57
398	Tokyo Collectables, Ltd	4	105548.73
406	Auto Canal+ Petit	3	86436.97
409	Stuttgart Collectable Exchange	0	0.00
412	Extreme Desk Decorations, Ltd	3	90332.38
415	Bavarian Collectables Imports, Co.	1	31310.09
424	Classic Legends Inc.	3	69214.33
443	Feuer Online Stores, Inc	0	0.00
447	Gift Ideas Corp.	3	49967.78
448	Scandinavian Gift Ideas	3	120943.53
450	The Sharp Gifts Warehouse	4	143536.27
452	Mini Auto Werke	3	51059.99

455	Super Scale Inc.	2	70378.65
456	Microscale Inc.	2	29230.43
458	Corrida Auto Replicas, Ltd	3	112440.09
459	Warburg Exchange	0	0.00
462	FunGiftIdeas.com	3	88627.49
465	Anton Designs, Ltd.	0	0.00
471	Australian Collectables, Ltd	3	55866.02
473	Frau da Collezione	2	25358.32
475	West Coast Collectables Co.	2	43748.72
477	Mit Vergnügen & Co.	0	0.00
480	Kremlin Collectables, Co.	0	0.00
481	Raanan Stores, Inc	0	0.00
484	Iberia Gift Imports, Corp.	2	50987.85
486	Motor Mint Distributors Inc.	3	77726.59
487	Signal Collectibles Ltd.	2	42570.37
489	Double Decker Gift Stores, Ltd	2	29586.15
495	Diecast Collectables	2	65541.74
496	Kelly's Gift Shop	4	137460.79

Best Baseball Players

Question

- *This question uses* `lahmansdb_midterm.batting` , `lahmansdb_midterm.pitching` *and* `lahmansdb_midterm.people` *. You previously loaded this information.*
- *There query computes performance metrics:*
  - *Batting:*
    - *On-base percentage: OBP is (sum(h) + sum(BB))/(sum(ab) + sum(BB)). This value is* `NULL` *if* `sum(ab) = 0` *.*
    - *Slugging percentage: SLG is defined by the function below. The value is* `NULL` *if* `sum(ab) = 0` *.*

```
(
(sum(h) - sum(`1b`) - sum(`2b`) - sum(`3b`) - sum(hr)) +
2*sum(`2b`) + 3*sum(`3b`) + 4*hr
)/sum(ab)
```

  - *Pitching:*
    - *total_wins is* `sum(w)` *.*

    - *total_loses is* `sum(l)`.
    - *win_percentage is* `sum(w)/(sum(w) + sum(l))`. *This value is NULL if* `sum(w) + sum(l) = 0`.

- *Professor Ferguson has two criteria for someone being a great baseball player. A play must meet at least one of the criteria to be a great baseball player.*
  - *Batting:*
    - *Total number of* `ab >= 1500`.
    - *SLG: Career* `SLG >= .575`
  - *Pitching:*
    - `(sum(w) + sum(l)) >= 200`.
    - `win_percentage >= 0.70)` *or* `sum(w) >= 300.`
- *In your result table there is some additional guidance.*
  - `great_because` *is either* `Pitcher` *or* `Batter` *based on whether the player matched the batting or pitching criteria.*
  - *The values from* `batting` *are* `None` *if the player did not qualify based on batting.*
  - *The values from* `pitching` *are* `None` *if the player did not qualify on pitching.*

Note: *For this query to run efficiently, you will need to create indexes on the tables.*

Answer

- *Execute your create index statements below.*

In [34]:
```
%%sql
# DROP INDEX index_batting;
CREATE INDEX index_batting
ON lahmansdb_midterm.batting (AB, H, 2B,3B, HR,BB);
#1B

# DROP INDEX index_pitching;
CREATE INDEX index_pitching
ON lahmansdb_midterm.pitching (W, L)
```

```
 * mysql+pymysql://root:***@localhost
0 rows affected.
0 rows affected.
```

Out[34]:
```
[]
```

- *Execute your SQL statement producing the query result below.*

In [35]:
```
%%sql
with career_basic as (
    select
        playerid,
        (sum(h) - sum(hr) - sum(`2b`) - sum(`3b`)) as career_singles,
        sum(`2b`) as career_doubles,
        sum(`3b`) as career_triples,
```

```
        sum(hr) as  career_hrs,
        sum(ab) as career_abs,
        sum(h) as career_hits,
        sum(bb) as career_walks
    from
        lahmansdb_midterm.batting
    group by playerid
),
    pi_summary as (
        select playerid,
               sum(w) as pi_win,
               sum(l) as pi_lose,
               sum(w)+sum(l) as pi_decision
        from
            lahmansdb_midterm.pitching
        group by playerid
    ),
    career_averages as (
        select
            playerid, career_abs, career_hits,
            career_singles, career_doubles, career_triples, career_hrs,
            career_walks,
            if(career_abs=0, NULL, (career_hits + career_walks)/(career_abs + career_w
            if(career_abs = 0,
                null,
               (career_singles + 2 * career_doubles + 3 * career_triples + 4 * career_
                ) as slg
        from career_basic
    ),
    pi_average as (
        select
            playerid, pi_win, pi_lose,pi_decision,
            if (pi_decision=0, NULL, ((pi_win)/pi_decision)) as win_percentage
            from pi_summary
    ),
    career_end as(
        select playerid,career_abs,career_singles,career_doubles,career_triples,career
        from career_averages
        having slg>0.575 and career_abs>=1500
    ),
    pi_end as (
        select playerid, pi_win, pi_lose, pi_decision, win_percentage
        from pi_average
        having pi_decision>=200 and (win_percentage>0.7 or pi_win>=300)
    )


SELECT playerid,career_abs,career_singles,career_doubles,career_triples,career_hrs,obp
FROM career_end
UNION
SELECT playerid,Null as career_abs,Null as career_singles,Null as career_doubles,Null
FROM pi_end;
```

```
 * mysql+pymysql://root:***@localhost
36 rows affected.
```

Out[35]:

playerid	career_abs	career_singles	career_doubles	career_triples	career_hrs	obp	slg	pi_wi
ruthba01	8398	1517	506	136	714	0.4718	0.6898	Non
hornsro01	8173	1919	541	169	301	0.4308	0.5765	Non
gehrilo01	8001	1531	534	163	493	0.4447	0.6324	Non
foxxji01	8134	1529	458	125	534	0.4275	0.6093	Non
greenha01	5193	847	379	71	331	0.4103	0.6050	Non
dimagjo01	6821	1333	389	131	361	0.3947	0.5788	Non
willite01	7706	1537	525	71	521	0.4806	0.6338	Non
bondsba01	9847	1495	601	77	762	0.4428	0.6069	Non
mcgwima01	6187	785	252	6	583	0.3922	0.5882	Non
ramirma02	8244	1452	547	20	555	0.4077	0.5854	Non
troutmi01	4656	792	268	49	310	0.4137	0.5831	Non
spaldal01	None	None	None	None	None	None	None	25
galvipu01	None	None	None	None	None	None	None	36
keefeti01	None	None	None	None	None	None	None	34
welchmi01	None	None	None	None	None	None	None	30
radboch01	None	None	None	None	None	None	None	31
clarkjo01	None	None	None	None	None	None	None	32
nichoki01	None	None	None	None	None	None	None	36
youngcy01	None	None	None	None	None	None	None	51
mathech01	None	None	None	None	None	None	None	37
planked01	None	None	None	None	None	None	None	32
johnswa01	None	None	None	None	None	None	None	41
alexape01	None	None	None	None	None	None	None	37
grovele01	None	None	None	None	None	None	None	30
wynnea01	None	None	None	None	None	None	None	30
spahnwa01	None	None	None	None	None	None	None	36
perryga01	None	None	None	None	None	None	None	31
niekrph01	None	None	None	None	None	None	None	31
carltst01	None	None	None	None	None	None	None	32
ryanno01	None	None	None	None	None	None	None	32
suttodo01	None	None	None	None	None	None	None	32
seaveto01	None	None	None	None	None	None	None	31
clemero02	None	None	None	None	None	None	None	35
maddugr01	None	None	None	None	None	None	None	35