Common Table Expressions Ranking, and Partitioning

Naming resultsets

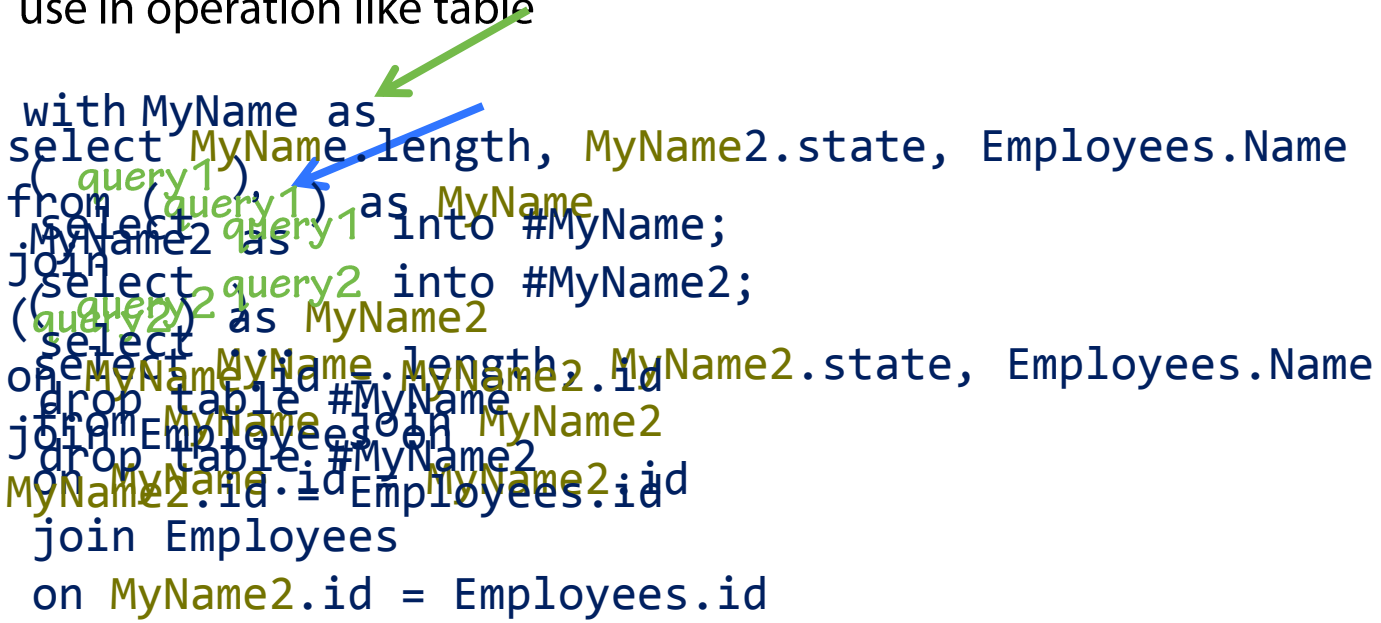
Where you at?

Overview

- **Common Table Expression**
  - named resultset common to expression
  - syntax only, not persisted
- **Top**
- **Ranking**
  - rows numerically ranked ← sort criterion
  - unique ≀ shared
- **Partition windows**
  - resultset partitioned into windows ← selection criterion
  - resultset processed *by window*

Common Table Expression

- **With clause precedes expression**
  - name
  - subquery-like syntax ı comma separated
  - use in operation like table



Top

- **Uses first N as ordered**
- **Traditional**
  - literal number ≀ select, order by
- **Current**
  - (expression) ≀ select, insert, update, and delete

name	price	weight
X4-gear	2.98	34
X9-gear	3.00	12
AY3-arbor	1.12	9
PR1-pin	32.99	18
PH9-pin	12.87	54
K4-shaft	4.56	32
K6-shaft	2.15	78
LO-lock	2.98	2
DT9-drift	14.30	34
PG1-pin	33.12	15

Ranking

- **Assigns a number to a row**
  - ranking function indicates numeric rank relative to other rows
  - ordered value used to calculate
  - may be unique, depending on ranking function used

ranked by price	name	price	weight	ranked by weight quartile
4	X4-gear	2.98	34	3
5	X9-gear	3.00	12	1
1	AY3-arbor	1.12	9	1
9	PR1-pin	32.99	18	2
7	PH9-pin	12.87	54	4
6	K4-shaft	4.56	32	2
2	K6-shaft	2.15	78	4
3	LO-lock	2.98	2	3
8	DT9-drift	14.30	34	1
10	PG1-pin	33.12	15	2

Row_Number

- **Ranks rows ordered by value calculated from row**
  - unique rank

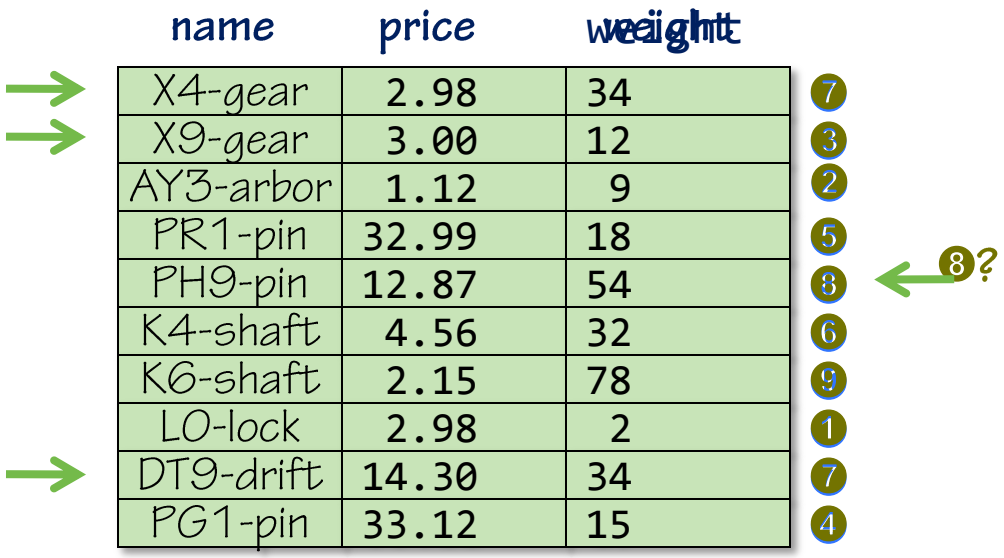
```
select name, row_number() over (order by      )
 from [mechanical parts];
```

	name	price	weight
4	X4-gear	2.98	34
5	X9-gear	3.00	12
1	AY3-arbor	1.12	9
9	PR1-pin	32.99	18
7	PH9-pin	12.87	54
6	K4-shaft	4.56	32
2	K6-shaft	2.15	78
3	LO-lock	2.98	2
8	DT9-drift	14.30	34
10	PG1-pin	33.12	15

Dense_Rank

- **Ranks rows ordered by value calculated from row**
  - same value means same rank
  - contiguous ← dense

```
select name, dense_rank() over(order by weight)
 from [mechanical parts];
```



NTile

- **Ranks rows by tile membership**
  - resultset ordered, broken into about equal size sequences, *i.e.* tiles
  - ranked as member of 1st, 2nd, 3rd, 4th *etc.* tile

quartiles

```
select name, ntile(4) over (order by weight)
  from [mechanical parts]
```

name	price	weight	
X4-gear	2.98	34	3
X9-gear	3.00	12	1
AY3-arbor	1.12	9	1
FR1-pin	32.99	18	2
PH9-pin	12.87	54	4
K4-shaft	4.56	32	2
K6-shaft	2.15	78	4
LO-lock	2.98	2	3
DT9-drift	14.30	34	1
FG1-pin	33.12	15	2

Partitioning

- **Ranking function applied by partition**
  - logical window ≀ defined by predicate ≀ applied to resultset
  - default is entire resultset

defines windows

```
select name,
 rank() over (partition by right(name, len(name) - charindex('-', name))
 order by weight) from [mechanical parts]
```

	name	price	weight	
gear partition / gear class	X4-gear	2.98	34	1
	X9-gear	3.00	12	2
arbor partition	AY3-arbor	1.12	9	1
pin partition	PR1-pin	32.99	18	2
	PH9-pin	12.87	54	1
shaft partition	K4-shaft	4.56	32	2
	K6-shaft	2.15	78	1
lock partition	LO-lock	2.98	2	1
drift partition	DT9-drift	14.30	34	1
	PG1-pin	33.12	15	3

Aggregate Partitions

- **Aggregates can be applied to partitions**
  - no column restrictions
  - mixed partitions

```
select right(name, len(name) - charindex('-', name)),
sum(price)over (partition by right(name, len(name) - charindex('-', name)))
from [mechanical parts]
```

aggregate

window

arbor	1.12
drift	14.30
gear	5.98
gear	5.98
lock	2.98
pin	78.99
pin	78.99
pin	78.99
shaft	6.71
shaft	6.71

name	price	weight
X4-gear	2.98	34
X9-gear	3.00	12
AY3-arbor	1.12	9
PR1-pin	32.99	18
PH9-pin	12.87	54
K4-shaft	4.56	32
K6-shaft	2.15	78
LO-lock	2.98	2
DT9-drift	14.30	34
PG1-pin	33.12	15

Summary

- CTE's are an alternate syntax for subqueries
- Ranking produces rank according to order by
  - row_number, dense_rank, ntile
- Ranking can be partitioned
- Aggregates can be partitioned

References