Data Advanced

H2. Database objects

Koen Bloemen

Sander De Puydt

DE HOGESCHOOL MET HET NETWERK

Elfde-Liniestraat 24, 3500 Hasselt, www.pxl.be

H2. Database Objects

Iets dat data vasthoudt en kan manipuleren:

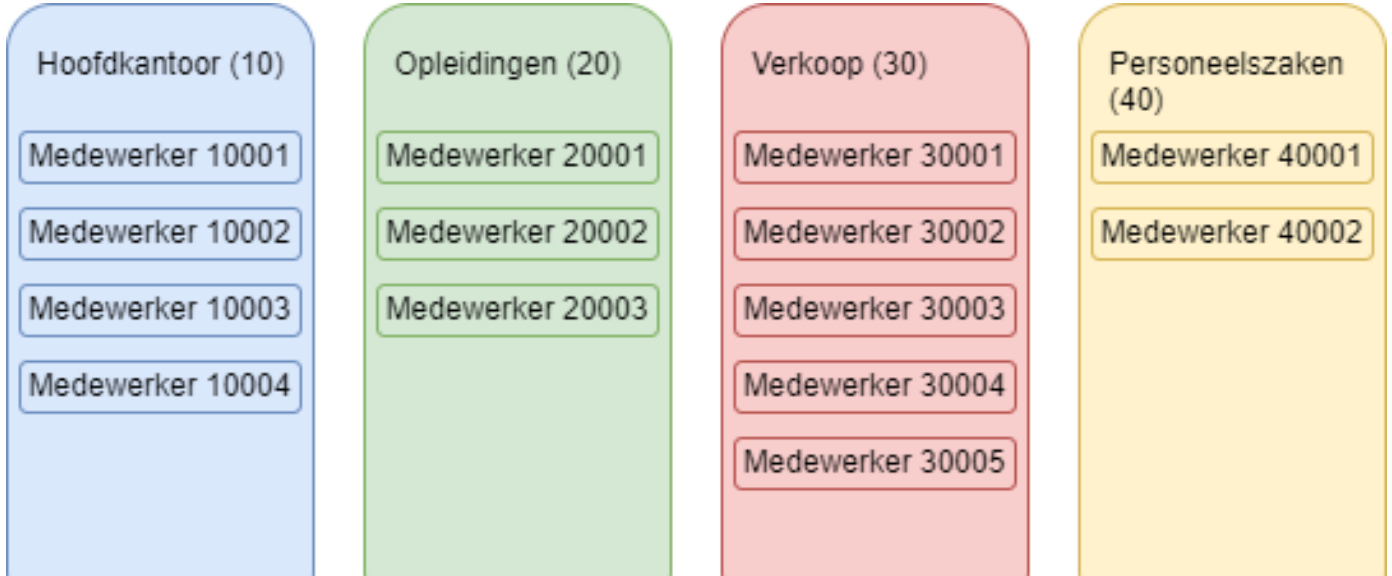
- Sequence
- Index
- Synonym
- ... (Er zijn er nog veel meer)

Sequence

- Doel
- CREATE Syntax
- ALTER Syntax
- DROP Syntax
- Testing

Sequence – Doel

- Use case: We willen een patroon gebruiken
- We willen de medewerkers codes geven op basis van hun afdeling. Elk van de medewerkers hebben unieke codes nodig die op elkaar opvolgen. Hoe zorgen we er voor dat we deze codes kunnen oproepen zonder de tabellen te moeten uitlezen (en zonder conflicten)?



Sequence – Doel

- Met een Sequence kunnen we er voor zorgen dat elke rij een unieke waarde heeft.
- We kunnen zelf bepalen wat het patroon is dat de Sequence moet volgen.
  - Dalend / Stijgend
  - In sprongen
  - Min / Max
- We kunnen inzicht krijgen in het verloop van de sequence tijdens het gebruik ervan door de huidige waarde op te vragen.
- Sequences kunnen gebruikt worden onafhankelijk van een tabel

Sequence – CREATE Syntax

CREATE SEQUENCE sequence_name
START WITH initial_value
INCREMENT BY increment_value
MINVALUE minimum value
MAXVALUE maximum value
CYCLE|NOCYCLE ;

Sequence – CREATE Syntax

START WITH initial_value

- Eerste waarde die gegenereerd wordt door de sequence
- Let op: Is niet de waarde waar de cyclus naar toe gaat, indien de sequence herhaald wordt.

INCREMENT BY increment_value

- Bepaalt het interval tussen twee getallen in een sequence
- Default = 1

Sequence – CREATE Syntax

MINVALUE minimum value

- *MINVALUE* moet kleiner of gelijk zijn aan *START WITH*
- *MINVALUE* moet kleiner zijn dan *MAXVALUE*

MAXVALUE maximum value

- *MAXVALUE* moet groter of gelijk zijn aan *START WITH*
- *MAXVALUE* moet groter zijn dan *MINVALUE*

CYCLE|NOCYCLE ;

- *CYCLE:* geeft aan dat de sequence getallen blijft genereren na het bereiken van een limiet
- *Default = NOCYCLE*

Sequence – CREATE Syntax

Voorbeeld:

```
CREATE SEQUENCE numbers_seq
START WITH 1
INCREMENT BY 1;
```

Sequence – ALTER Syntax

ALTER SEQUENCE sequence_name

START WITH initial_value

INCREMENT BY increment_value

MINVALUE minimum value

MAXVALUE maximum value

RESTART

CYCLE|NOCYCLE ;

Sequence – ALTER Syntax

MINVALUE minimum value

MAXVALUE maximum value

- Oracle controleert of een nieuwe MINVALUE of MAXVALUE in conflict is met het huidge sequence number

RESTART

- Gebruik RESTART om de NEXTVAL in te stellen op MINVALUE voor een ascending sequence en op MAXVALUE voor een descending sequence.
- Indien je wil herstarten op een ander getal, dan **combineer** je RESTART met START WITH

Sequence – ALTER Syntax

Voorbeeld:

```
ALTER SEQUENCE numbers_seq
START WITH 100
INCREMENT BY -1
MINVALUE 0
RESTART
CYCLE ;
```

Sequence – DROP Syntax

DROP SEQUENCE sequence_name;

Voorbeeld:

DROP SEQUENCE numbers_seq;

Sequence – Testing

- Hoe kunnen we onze sequence uitproberen?
  - DUAL tabel
    - De DUAL tabel zit standard in je Oracle Database en kan gebruikt worden door elke USER.
    - De DUAL tabel heeft één kolom, DUMMY, met één rij met de waarde “X”.
    - De DUAL tabel wordt gebruikt om constante expressions met het SELECT statement uit te voeren. Aangezien er maar één rij is, wordt de constante waarde slechts één keer teruggegeven. *(Probeer het uit!)*
  - nextval
  - currval

```
SQL> SELECT * FROM DUAL;

D
-
X
```

Sequence – Testing

- Hoe kunnen we onze sequence uitproberen?
  - DUAL tabel
  - nextval
    - Geeft de eerstvolgende waarde in de sequence terug.
    - Elke keer je de **nextval** oproept wordt de **current value** van de sequence aangepast (ongeacht of je de waarde gebruikt hebt in een insert statement).
  - currval
    - Geeft de huidige waarde terug van de sequence.

Voorbeeld:

SELECT numbers_seq.currval FROM DUAL;

```
SQL> SELECT numbers_seq.currval FROM DUAL;

   CURRVAL
----------
        53
```

Index

- Doel
- CREATE Syntax
- ALTER Syntax
- DROP Syntax
- Testing
- Extra

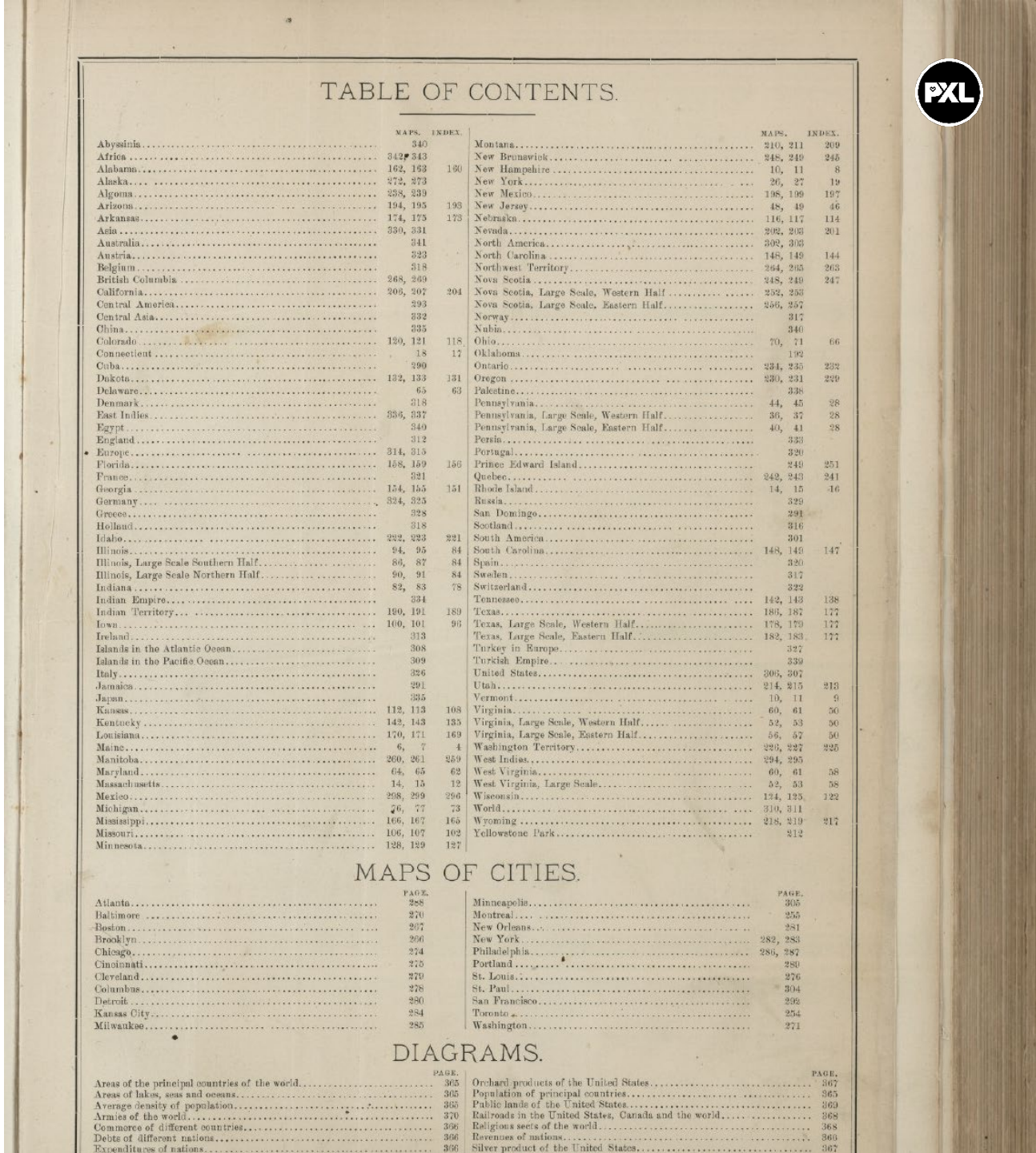


TABLE OF CONTENTS.

MAPS OF CITIES.

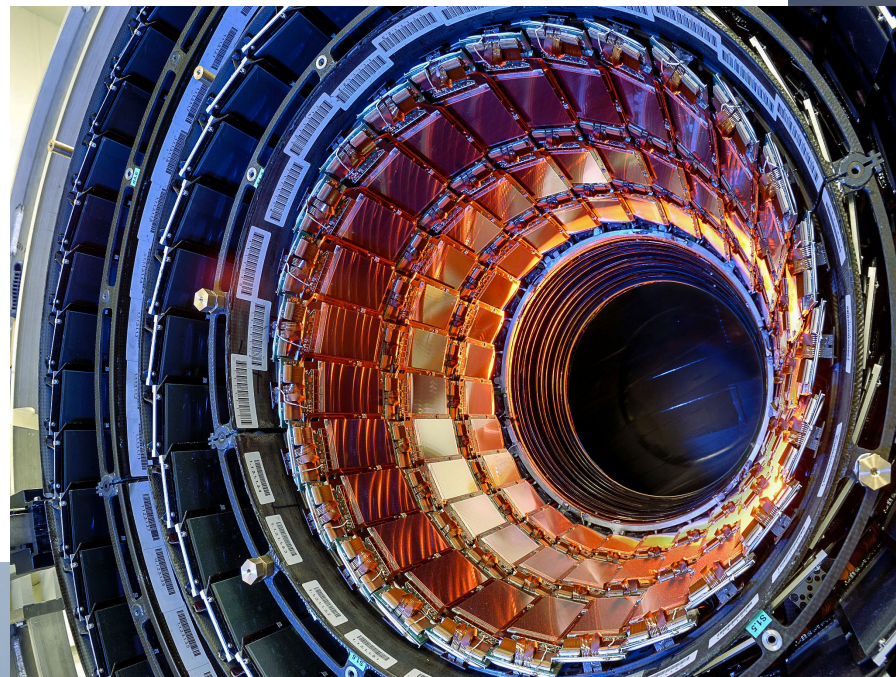
DIAGRAMS.

Index – Doel

- Use case:
- Ik heb honderdduizenden films en wil weten **welke + hoeveel** films zijn **uitgekomen in 1994** en oorspronkelijk in het **Frans** zijn geproduceerd en ik wil ze **gesorteerd** zien in de ranking van IMDB.
- Grote datasets, zoals IMDB, eisen veel tijd voor een query.
  - We kunnen de query sneller maken met een Index

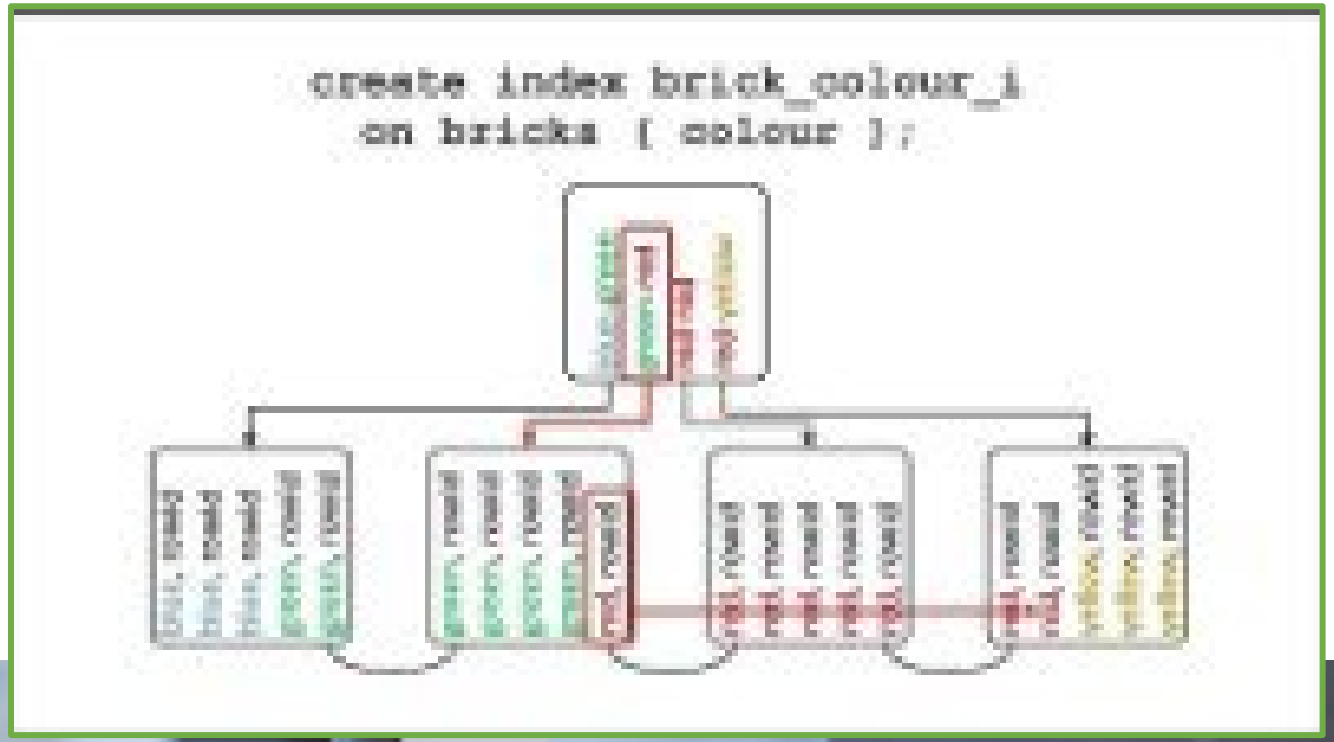
Index – Doel

- Databases kunnen bijzonder groot worden.
  - Large Hadron Collier in CERN genereert gigantische hoeveelheden aan data
  - 10^9 collisions/s x 1 Mbyte/collision = 10^{15} bytes/s = 1 PB/s (1 Petabyte/second)
- Hoe groter de database, hoe langer de zoektijd van een query wordt.
- Met een Index kunnen we onze database performant maken.



Index – Doel

- Duidelijke video over werking van indexen in databases
  - Geeft meer uitleg over multi-kolom indexen
  - Extra: geeft uitleg over function-indexen
- Video **voor thuis** →



Index – Doel

- Hoe werkt een index?
- Zonder index de naam 'Zack' opzoeken in een tabel.
  - Eén per één vergelijken
  - Overloopt volledige tabel
  - (full scan)
  - Traag

SELECT * FROM friends WHERE name = 'Zack';

friends		
id	**name**	**city**
1	Matt	San Francisco
2	Dave	Oakland
3	Andrew	Blacksburg
4	Todd	Chicago
5	Blake	Atlanta
6	Evan	Detroit
7	Nick	New York City
8	Zack	Seattle

Index – Doel

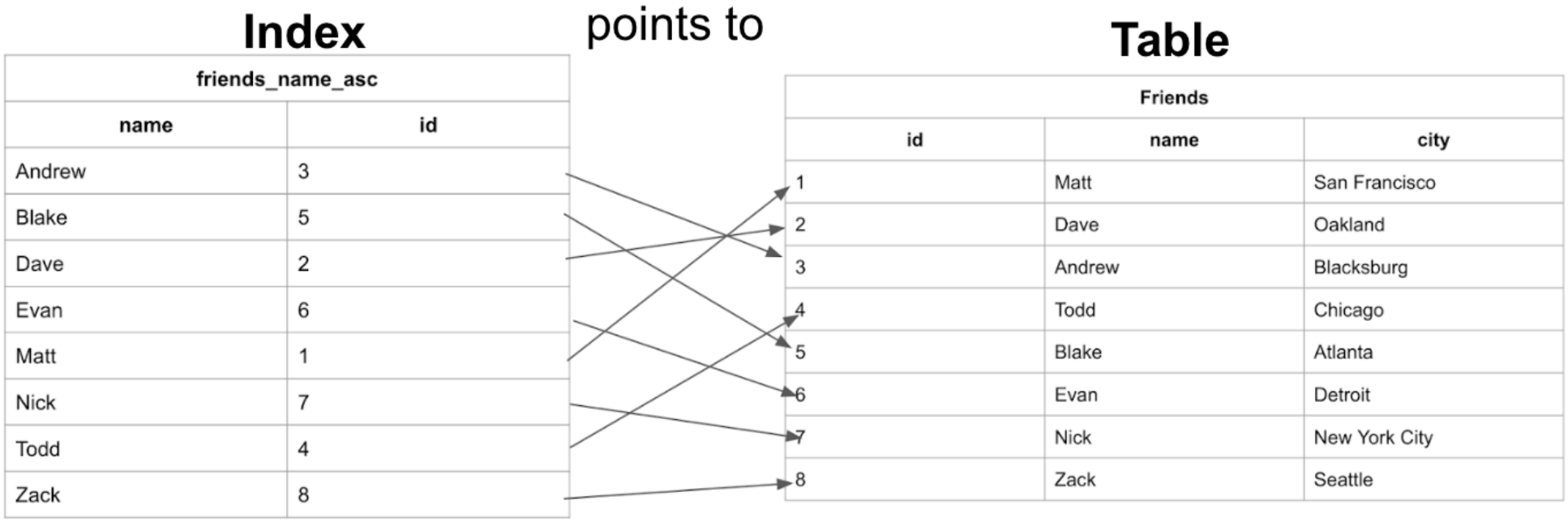
- Hoe werkt een index?
- Met index de naam 'Zack' opzoeken in een tabel.
  - Slaat veel records over
  - Geen full scan
  - Snel
  - … Maar hoe?

SELECT * FROM friends WHERE name = 'Zack';

friends_name_asc	
Name	**Index**
Andrew	3
Blake	5
Dave	2
Evan	6
Matt	1
Nick	7
Todd	4
Zack	8

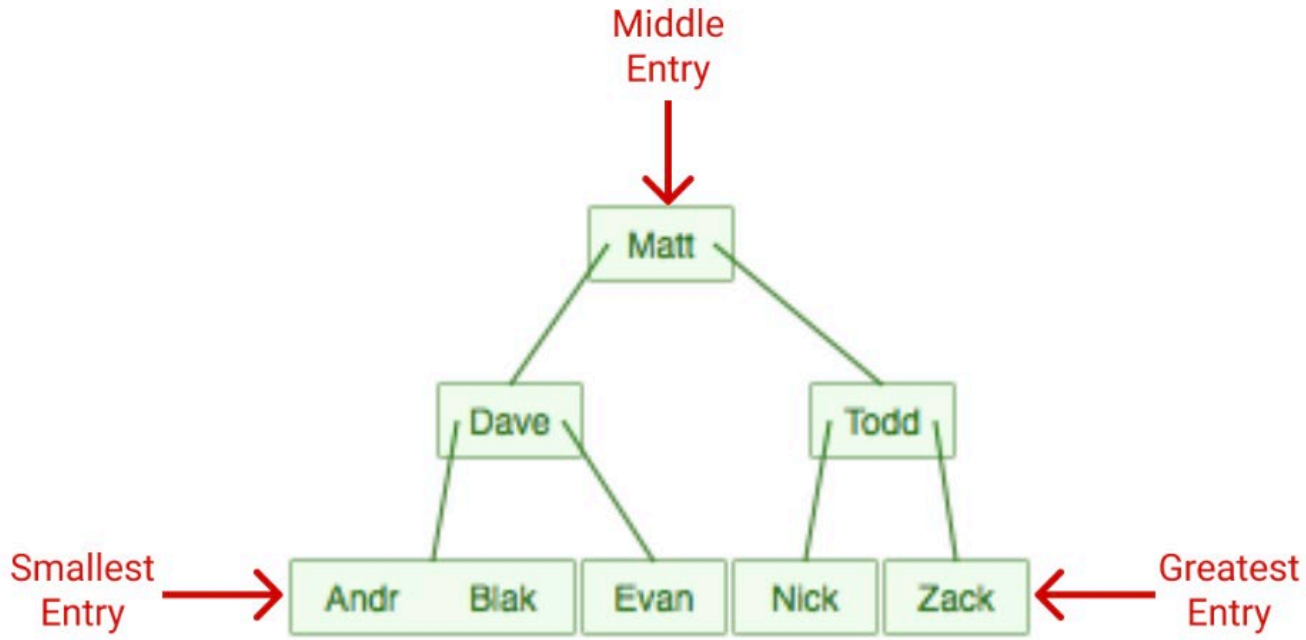
Index – Doel

- Hoe werkt een index?
- Data wordt gestructureerd rond de geïndexeerde kolom



Index – Doel

- Hoe werkt een index?
- Het doorlopen van deze data gebruikt een boomstructuur



Index – CREATE Syntax

- Het aanmaken van een index is "makkelijk"
  - Syntax = makkelijk
- Het kiezen van de index strategie is "moeilijk"
  - Strategie = moeilijk

CREATE INDEX index_name
ON table_name
(kolom_naam1, kolom_naam2, ...);

Index – CREATE Syntax

- De index maak je aan op basis van een kolom. De index kan een query versnellen, **als** de query een vraag stelt over de geïndexeerde kolom.
  - Je kan meerdere indexes maken per tabel, maar
  - Hoe meer indexes, hoe trager **inserts en deletes zijn (die rekening moeten houden met het onderhoud van de index)**
  - *CREATE INDEX index_name ON table_name (kolom_naam1);*
- Wanneer je meer complexe queries maakt die condities hebben voor meerdere kolommen, dan kan je een multi-kolom index maken
  - Meerdere index =! Multi-kolom index
  - Volgorde van de kolommen in CREATE bepaalt de volgorde waarin gezocht wordt
  - *CREATE INDEX index_name ON table_name (kolom_naam1, kolom_naam2, ...);*

Index – CREATE Syntax

- Samengevat:
  - Multiple indexes kan slecht zijn voor database performantie.
    - Wees voorzichtig met het aanmaken hiervan
  - Multi-kolom index is beter, maar je moet op voorhand goed weten welke queries vaak uitgevoerd worden.
    - Je index kan enkel specifieke queries op de gegeven kolommen versnellen.

Index – ALTER Syntax

- Past een bestaande index aan:

ALTER INDEX index_name
ON table_name
(kolom_naam1, kolom_naam2, …);

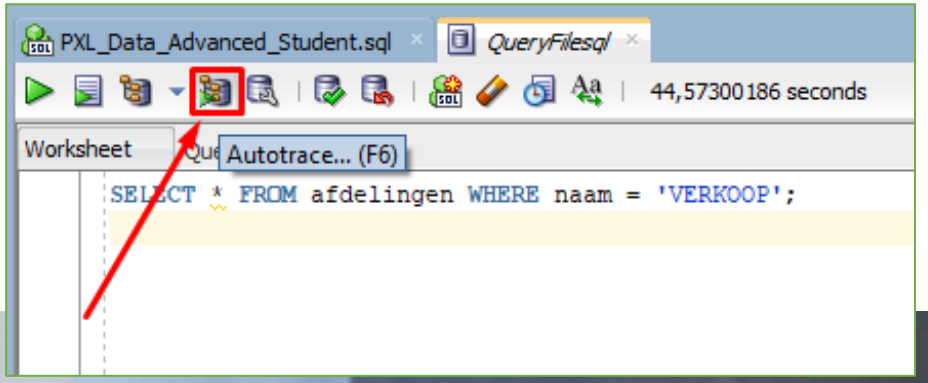
Index – DROP Syntax

- Verwijdert een bestaande index:

DROP INDEX index_name;

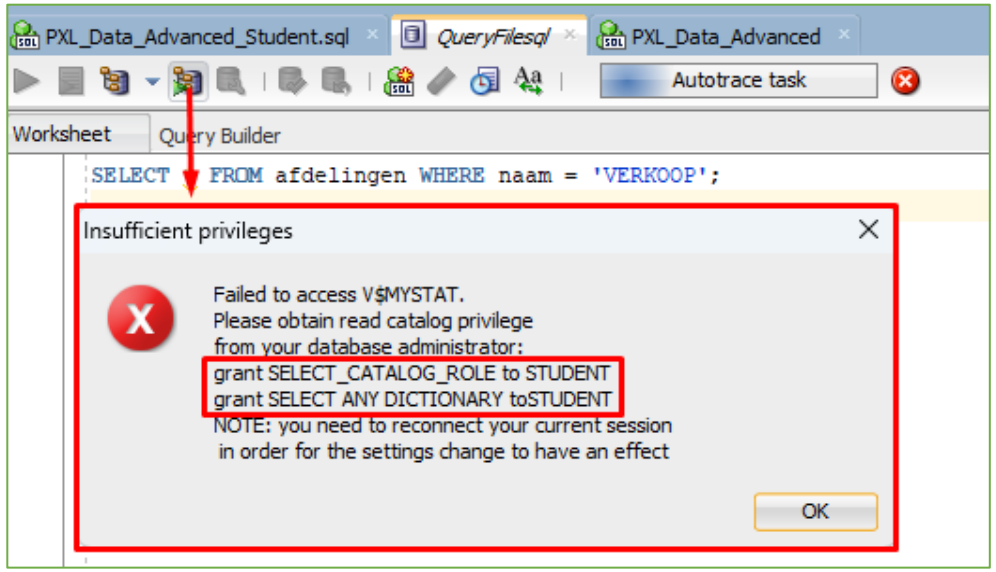
Index – Testing

- Wanneer we aan SQL optimalisatie doen, hoe kunnen we testen of onze optimalisatie succesvol is.
  - Achterliggend wordt er een EXECUTION PLAN gebruikt
  - Dit plan bepaalt welke tabellen worden aangesproken en in welke volgorde
  - Controleer de SCAN OPTIONS
- We kunnen dit plan voor een bepaalde query terugvinden in SQL Developer onder "Autotrace"

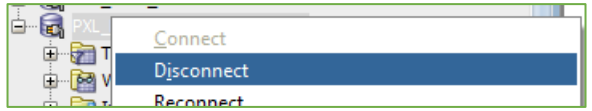


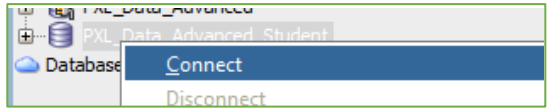
Index – Testing

- Wanneer je Autotrace activeert, dan kan het zijn dat je nog rechten mist. Voeg deze toe als "system" user + Reconnect.



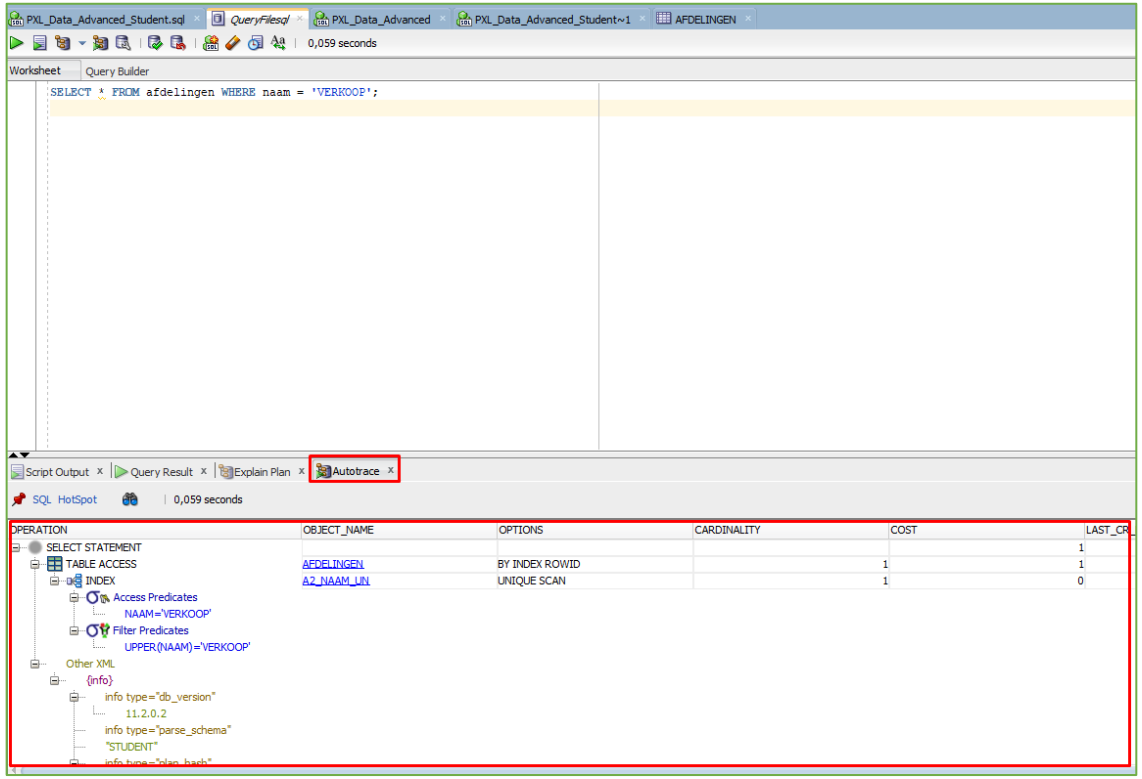

```
SQL> connect system/pxl
Connected.
SQL> grant SELECT_CATALOG_ROLE to student;

Grant succeeded.

SQL> grant SELECT ANY DICTIONARY to student;

Grant succeeded.
```





Index – Testing

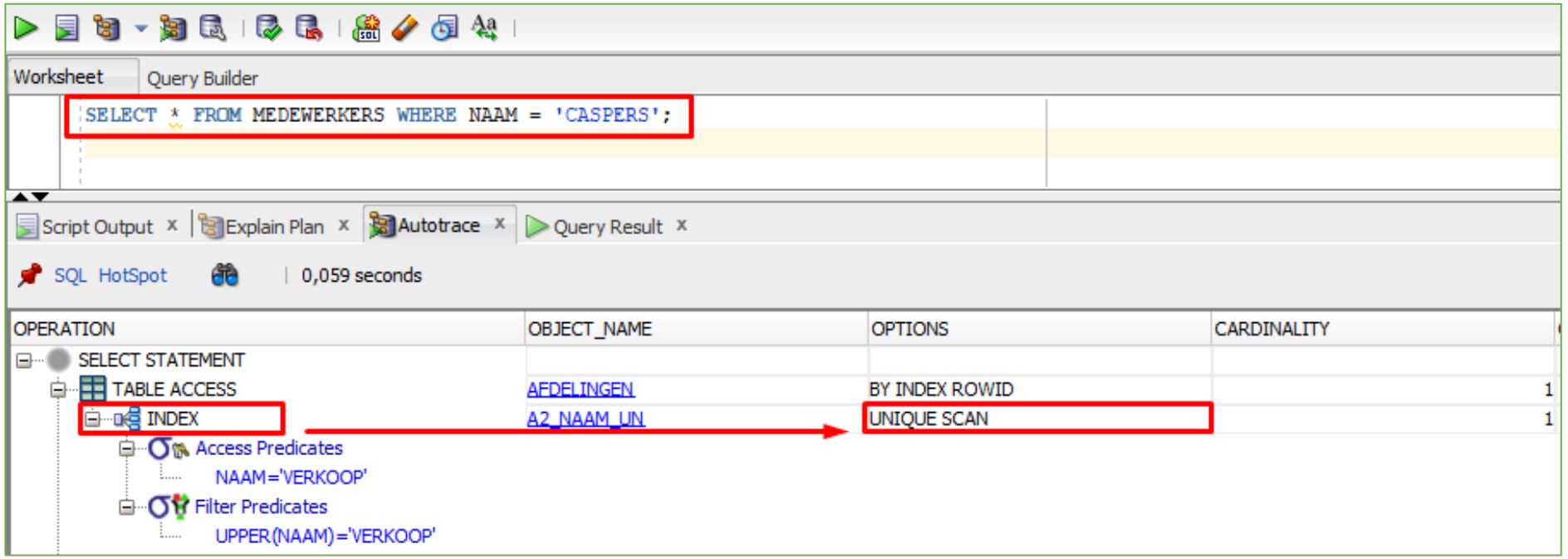
- Autotrace



Hoe kunnen we het gebruik van onze index interpreteren?

Index – Testing

- Autotrace: Geef een query in op de MEDEWERKERS tabel
  - Er wordt een index unique scan gebruikt.



Index – Testing

- Autotrace: Geef een query in op de MEDEWERKERS tabel
  - Er wordt een index unique scan gebruikt.
  - Maar … we hebben nooit een index aangemaakt voor MEDEWERKERS ?!

  - **Let op: Oracle maakt een index automatisch aan wanneer een unique key of primary key wordt gebruikt!**

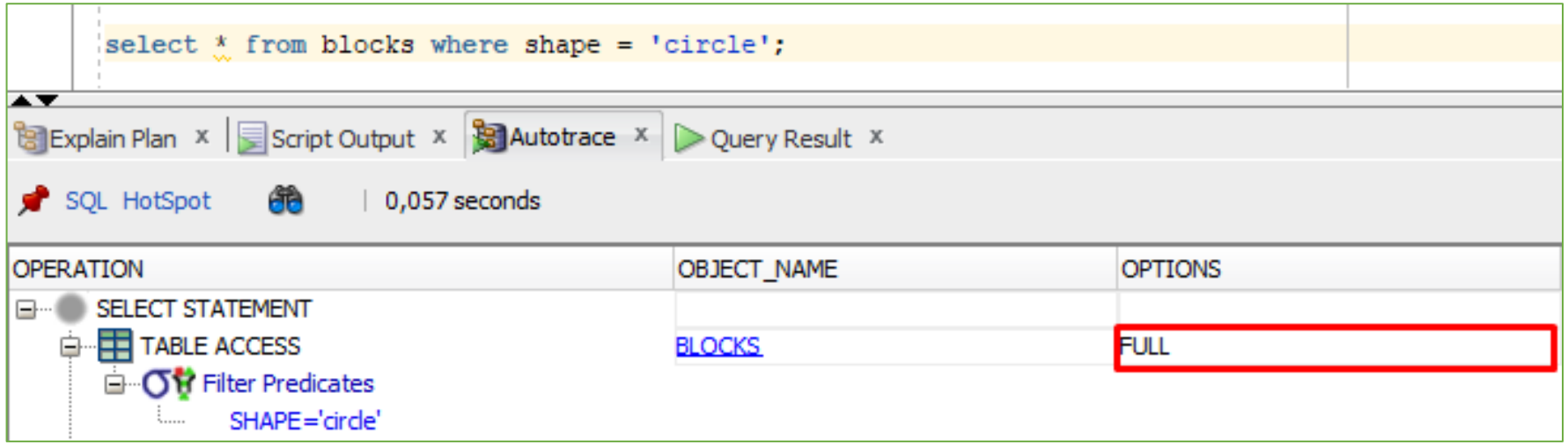
Index – Testing

- Voorbeeld zonder unique primary key constraint

```
/*
CREATE TABLE Blocks (BlockID int, Color varchar(20), Shape varchar(20));
INSERT INTO blocks VALUES (1, 'red', 'square');
INSERT INTO blocks VALUES (2, 'red', 'triangle');
INSERT INTO blocks VALUES (3, 'red', 'square');
INSERT INTO blocks VALUES (4, 'blue', 'square');
INSERT INTO blocks VALUES (5, 'blue', 'circle');
INSERT INTO blocks VALUES (6, 'blue', 'square');
INSERT INTO blocks VALUES (7, 'yellow', 'circle');
INSERT INTO blocks VALUES (8, 'blue', 'square');
INSERT INTO blocks VALUES (9, 'red', 'triangle');
*/

select * from blocks where shape = 'circle';
```

Index – Testing

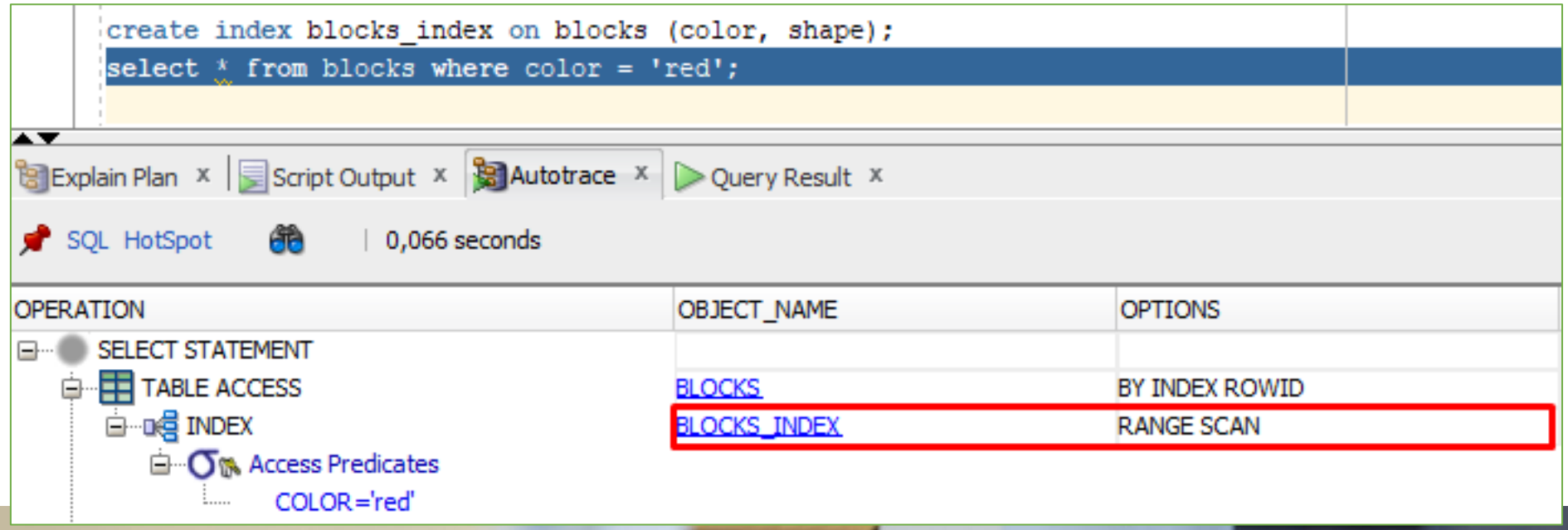
- Voorbeeld zonder unique primary key constraint
  - Er wordt **FULL SCAN** uitgevoerd
  - Er is geen index gebruikt



Index – Testing

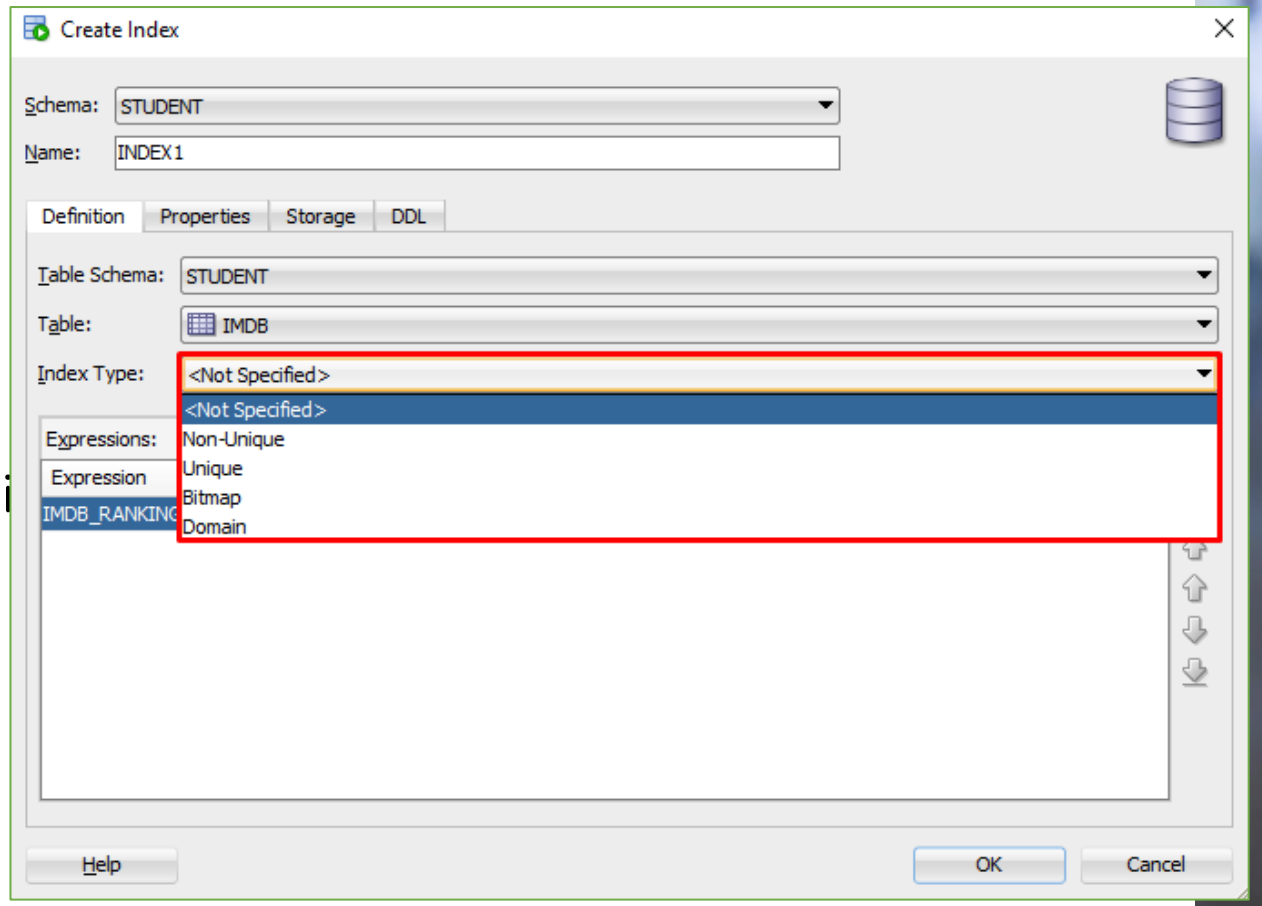
- Voorbeeld na aanmaken van index

```
create index blocks_index on blocks (color, shape);
select * from blocks where color = 'red';
```



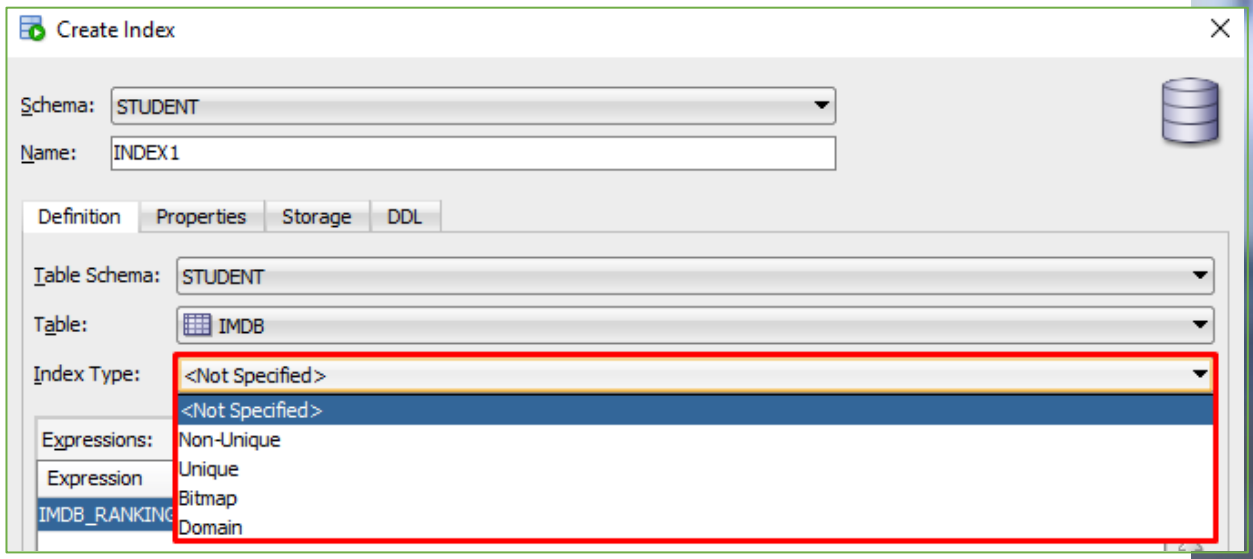
Index – Extra

- Als je SQL Developer gebruikt om een Index aan te maken, dan zal je zien dat je kan kiezen voor een type.
  - Unique:
    - Elke waarde in de kolom(men) i **uniek**
  - Non-Unique: default



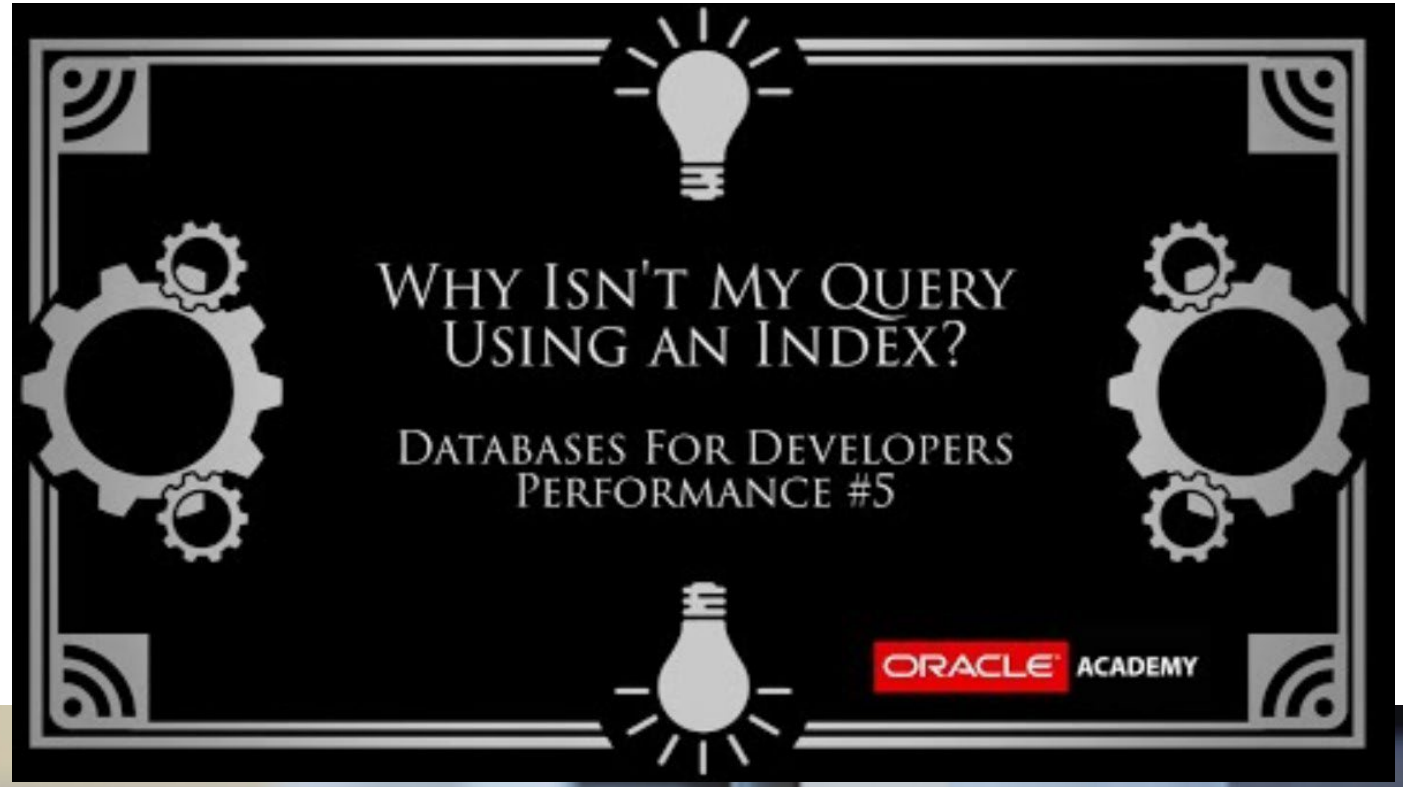
Index – Extra

- Als je SQL Developer gebruikt om een Index aan te maken, dan zal je zien dat je kan kiezen voor een type.
  - Het gebruik van de alternatieve types (bitmap en domain) vallen buiten de scope van dit vak:
    - **Bitmap**: beschrijft een **range van rijen** en slaat op welke rijen in de range wel of niet een waarde hebben in de kolom in een bitmap (voor elke mogelijke waarde van de kolommen).
    - Bitmaps zijn enkel goed voor datasets die nooit aangepast worden (data warehouse), aangezien het **write concurrency moet blokkeren** om **dirty writes** te voorkomen
    - **Domain**: specialiseerd in indexeren van **ruimtelijke data** of **afbeeldingen**



Index – Extra (niet kennen)

- Er zijn uitzonderingen waarin er toch geen index wordt gebruikt.
  - Ontdek waarom:



Synonym

- Doel
- CREATE Syntax
- ALTER Syntax
- DROP Syntax

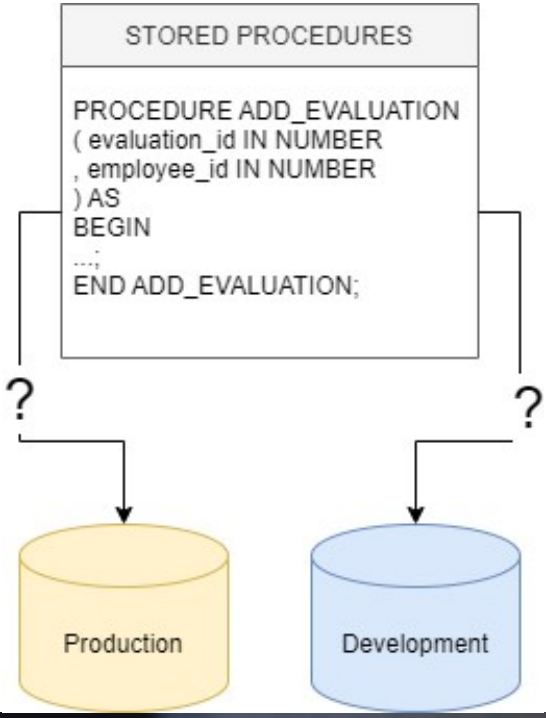
GOOD

-

excellent, fine, superior, wonderful, marvelous, qualified, suited, suitable, apt, proper, capable, generous, kindly, friendly, pleasant, satisfactory, well-behaved, obedient, honorable, reliable, top-notch, worthy, grand, salubrious, noble, great, beneficial, splendid, first-rate, genuine, ample, estimable, valid, helpful, expedient, righteous, advantageous, sterling, superb, respectable, edifying, trustworthy , gracious, obliging, agreeable, safe, profitable, beneficial

Synonym – Doel

- Use case:
- Er is een database object met een lange of onduidelijke naam. Hoe kunnen we een alternatieve naam introduceren?
- We hebben stored procedures (sql opdrachten) die gebruik maken van de productie database (met *prod_tables*), maar we willen graag testen uitvoeren op de development database (met *dev_tables*) zonder alle queries in de stored procedures aan te moeten passen.
  Hoe lossen we dit op?
  - **= Abstractie laag om productie applicaties te beschermen.**

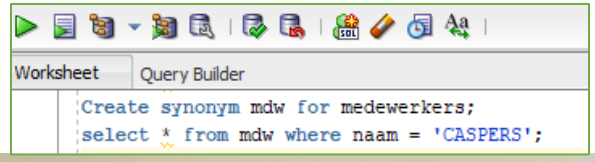


Synonym – CREATE Syntax

CREATE SYNONYM synonym_voor_data_object

FOR naam_van_data_object;

- *Indien je geen rechten hebt om een synonym aan te maken, dan zal je deze als system user moeten toevoegen aan student.*
  - Zie Prerequisites
  - Test een synonym uit!




```
SQL> connect system/pxl
Connected.
SQL> grant create synonym to student;

Grant succeeded.

SQL> grant create any synonym to student;

Grant succeeded.
```

Synonym – ALTER Syntax

ALTER SYNONYM synonym_voor_data_object
FOR naam_van_data_object COMPILE;

- COMPILE:
  - Een synonym plaatst een afhankelijkheid op het target data object en wordt onbruikbaar (*invalid*) wanneer het target data object aangepast of verwijderd wordt.
  - Wanneer je een onbruikbaar (*invalid*) synonym compiled, dan wordt het terug bruikbaar (*valid*).

Synonym – DROP Syntax

- Verwijdert een bestaande synonym:

DROP SYNONYM synonym_voor_data_object

Oefeningen!