



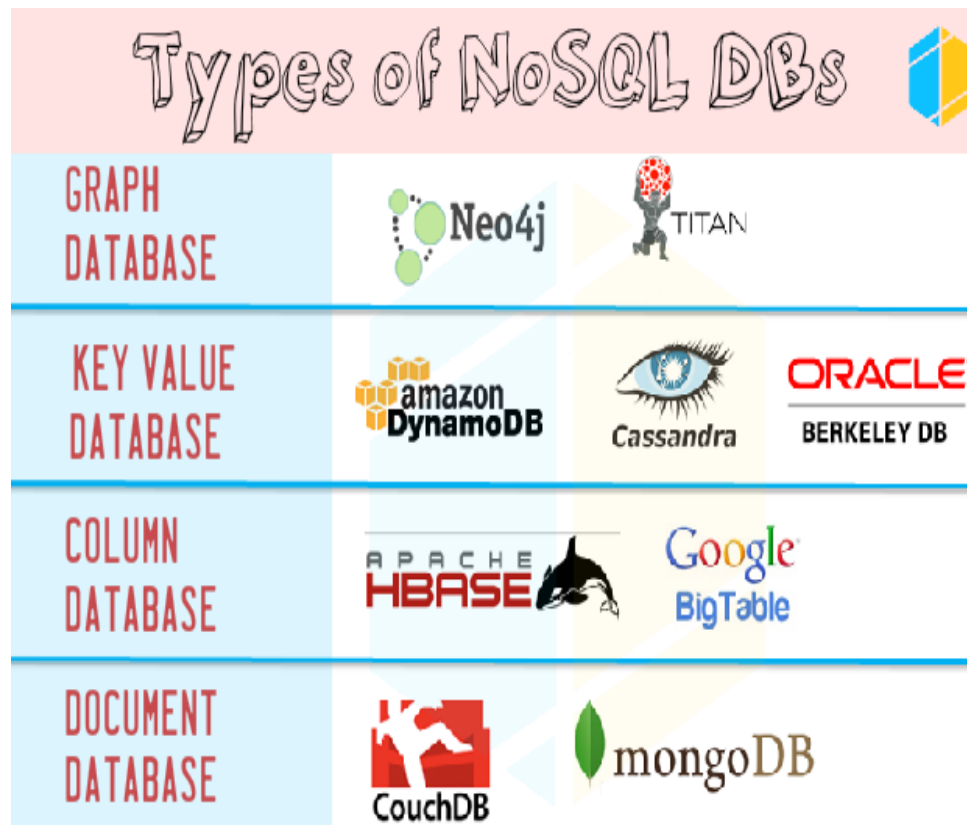
Modern Database Systems

Practicals. Multi-model databases

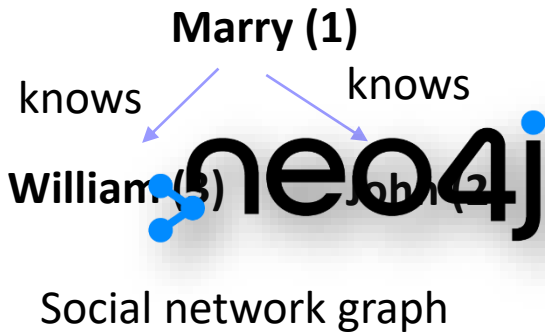
Doc. RNDr. Irena Holubova, Ph.D.

holubova@ksi.mff.cuni.cz

NoSQL Database Types

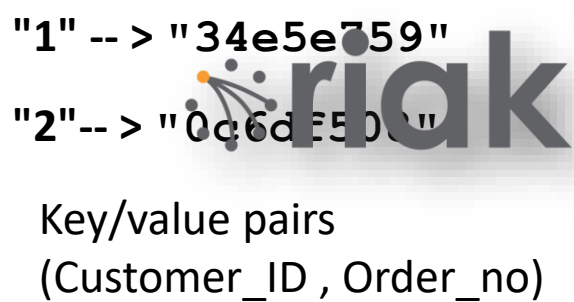


An example of multi-model data



```
{ "Order_no": "0c6df508",
  "Orderlines": [
    { "Product_no": "2724f",
      "Product_Name": "Toy",
      "Price": 66 },
    { "Product_no": "3424g",
      "Product_Name": "Book",
      "Price": 40 } ]
}
```

MongoDB



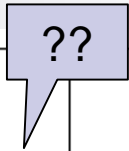
Order JSON doc

Customer relation

Order_ID	Name	Credit_limits
1	Mary	5,000
2	John	3,000
3	William	2,000

Overview of Supported Data Models

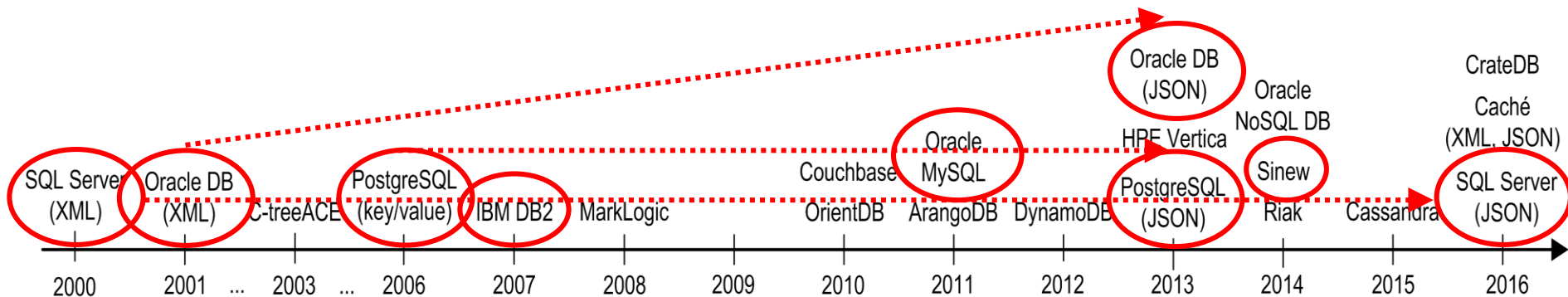
Type	DBMS	Relational	Column	Key/value	Document (JSON)	XML	Graph	Nested data/UDT/object
Relational	PostgreSQL	✓		✓	✓	✓		
	SQL Server	✓			✓	✓		
	IBM DB2	✓				✓		
	Oracle DB	✓			✓	✓		
	Oracle MySQL	✓		✓				
	Sinew	✓		✓				
Column	Cassandra		✓					✓
	CrateDB	✓	✓		✓			
	DynamoDB		✓	✓	✓			
	HPE Vertica		✓		✓			
Key/value	Riak			✓	✓	✓		
	c-treeACE	✓		✓				
	Oracle NoSQL DB	✓		✓				
Document	ArangoDB			✓	✓		✓	
	Couchbase			✓	✓			
	MongoDB			✓	✓		✓	
	Cosmos DB		✓	✓	✓		✓	
	MarkLogic				✓	✓		
Graph	OrientDB			✓	✓		✓	✓
Object	Caché	✓			✓	✓		✓



Relational Multi-model DBMSs

- Biggest set of multi-model databases
 - The most popular type of databases
 - SQL has been extended towards other data formats (e.g, SQL/XML)
 - Simplicity and universality of the relational model

Type	DBMS	Relational	Column	Key/value	Document (JSON)	XML	Graph	Nested data/UDT/object
Relational	PostgreSQL	✓		✓	✓	✓		
	SQL Server	✓			✓	✓		
	IBM DB2	✓				✓		
	Oracle DB	✓			✓	✓		
	Oracle MySQL	✓		✓				
	Sinew	✓		✓				





Relational Multi-model DBMSs

Storage – PostgreSQL Example

```
CREATE TABLE customer (  
    id          INTEGER PRIMARY KEY,  
    name       VARCHAR(50),  
    address    VARCHAR(50),  
    orders     JSONB  
);  
  
INSERT INTO customer  
VALUES (1, 'Mary', 'Prague',  
    '{"Order_no":"0c6df508",  
     "Orderlines": [  
       {"Product_no":"2724f", "Product_Name":"Toy", "Price":66},  
       {"Product_no":"3424g", "Product_Name":"Book", "Price":40}]  
    }');  
  
INSERT INTO customer  
VALUES (2, 'John', 'Helsinki',  
    '{"Order_no":"0c6df511",  
     "Orderlines": [  
       { "Product_no":"2454f", "Product_Name":"Computer", "Price":34 } ]  
    }');
```

id	name	address	orders
integer	character varying (50)	character varying (50)	jsonb
1	Mary	Prague	{"Orderlines":[{"Price":66,"Product_Name":"Toy","Product_no":"2724f"},{"Price":40,"Product_Name":...
2	John	Helsinki	{"Orderlines":[{"Price":34,"Product_Name":"Computer","Product_no":"2454f"}],"Order_no":"0c6df511"}

Relational Multi-model DBMSs

Storage – PostgreSQL Example

```
SELECT json_build_object('id',id,'name',name,'orders',orders)
FROM customer;
```

json_build_object json
{"orders":{"Orderlines":[{"Price":66,"Product_Name":"Toy","Product_no":"2724f"},{"Price":40,"Product_Name":"Book","Product_no":"3...
{"orders":{"Orderlines":[{"Price":34,"Product_Name":"Computer","Product_no":"2454f"}],"Order_no":"0c6df511"},"id":2,"name":"John"}

```
SELECT jsonb_each(orders) FROM customer;
```

jsonb_each record
(Order_no,""0c6df508"")
(Orderlines,"[{""Price"": 66, ""Product_no"": ""2724f"", ""Product_Name"": ""To...
(Order_no,""0c6df511"")
(Orderlines,"[{""Price"": 34, ""Product_no"": ""2454f"", ""Product_Name"": ""Co...

```
SELECT jsonb_object_keys(orders) FROM customer;
```

jsonb_object_keys text
Order_no
Orderlines
Order_no
Orderlines

SQL Extensions and SQL-Like Languages

PostgreSQL Example (relational)

id integer	name character varying (50)	address character varying (50)	orders jsonb
1	Mary	Prague	{"Orderlines":[{"Price":66,"Product_Name":"Toy","Product_no":"2724f"},{"Price":40,"Product_Name":...
2	John	Helsinki	{"Orderlines":[{"Price":34,"Product_Name":"Computer","Product_no":"2454f"}],"Order_no":"0c6df511"}



```

{"Order_no": "0c6df508",
 "Orderlines": [
  { "Product_no": "2724f",
    "Product_Name": "Toy",
    "Price": 66 },
  { "Product_no": "3424g",
    "Product_Name": "Book",
    "Price": 40 } ]
}

```

```

SELECT name,
       orders->>'Order_no' as Order_no,
       orders#>' {Orderlines,1}' ->>'Product_Name' as
Product_Name
FROM customer
where orders->>'Order_no' <> '0c6df511';

```

name character varying (50)	order_no text	product_name text
Mary	0c6df508	Book

Assignment

- Chose your unique problem domain
 - E.g., the results of football matches of various teams
- For your selected problem domain, think about an application that uses multi-model features of PostgreSQL (or any other multi-model database). Create tables and store respective data so that you use at least 2 distinct models. Create at least 3 cross-model queries.
- Submit a script with respective commands for PostgreSQL + explanatory comments

References

- Documentation:

<https://www.postgresql.org/docs/9.6/static/index.html>

- JSON functions and operators

<https://www.postgresql.org/docs/9.6/static/functions-json.html>