

Anden undervisningsgang

Database

Denne undervisningsgang

- Lektier fra sidst
- Forskellige database systemer
- Den relationelle model
- Normalformer uddybet
- Mere end CRUD
- Nøgler låser data op
- Forbind data til ny sæt - teori
- Forbind MySQL Workbench til MAMP
- Gennemgang af større eksempel

Lektier fra sidst

Det der er produceret vises og forklares til underviseren og resten af klassen

Forskellige database systemer

Og enkelte eksempler på brugere heraf

Oracle Database

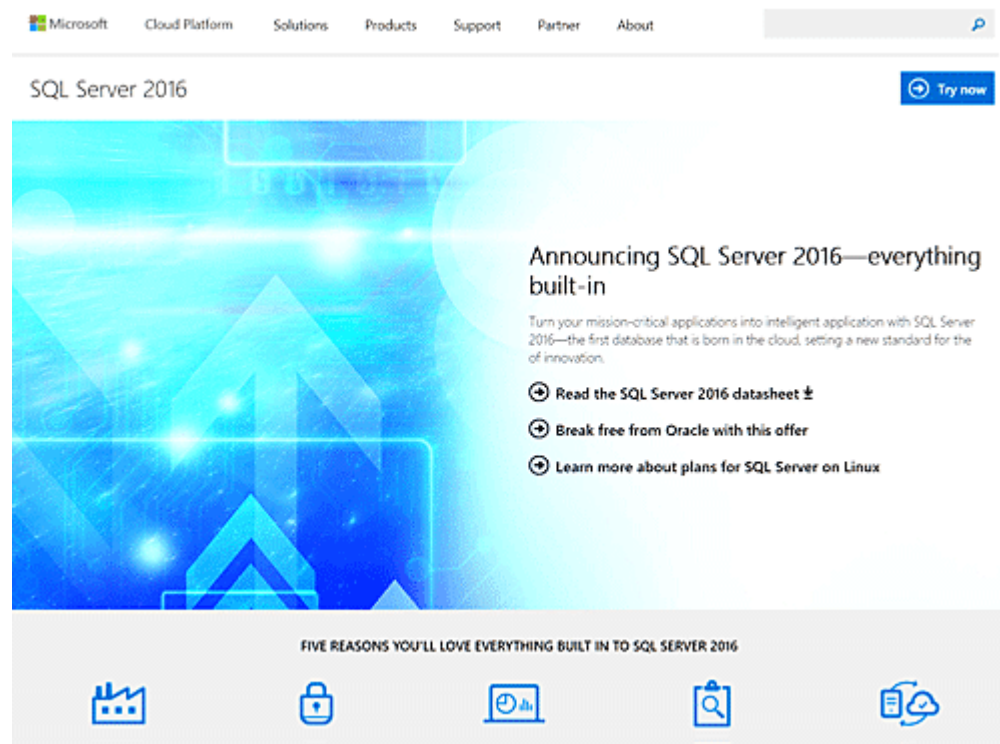


| The #1 Database Designed for the Cloud

Consolidate and manage databases as cloud services. Accelerate analytical performance—while achieving new levels of efficiency, security, and availability. Oracle Database can be rapidly provisioned and ready in minutes. With just a few clicks, you can stand up new database instances in a complete development environment. Best of all, there are no application changes when you move your legacy databases to Oracle's secure and optimized cloud platform.

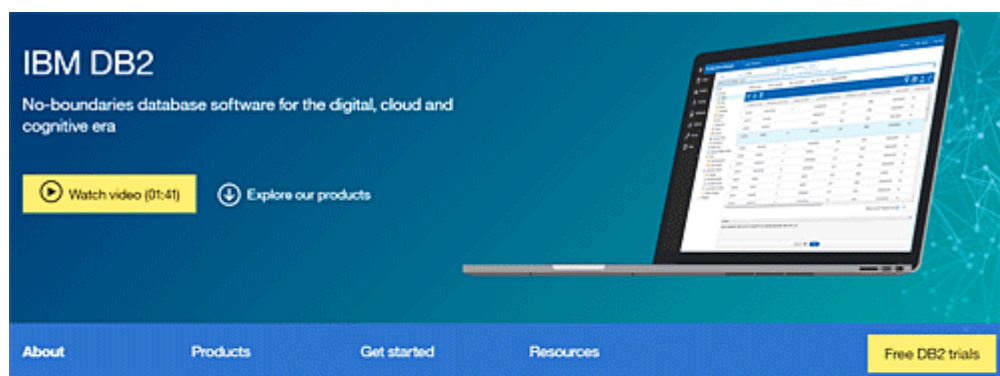
- Oracle begyndte i 1979 som det første kommercielt tilgængelige relationelle database management system (RDBMS).
- Oracle's navn er synonymt med enterprise database systemer, urokkelig data levering og hård virksomhedskonkurrence fra CEO Larry Ellison.
- Oracle Database understøtter standard [ANSI SQL](#) men har også deres egne udvidelser, f.eks. [Analytic SQL](#)

Microsoft SQL Server



- Hvis du kører en Microsoft Windows-baseret server, vil du sandsynligvis køre SQL Server på Microsoft SQL Server.
 - Kendt for sin brugervenlighed
- Understøtter standard SQL
- Har en række unikke værktøjer som SQL Server Profiler, SQL Server Management Studio, BI tools og Database Tuning Advisor.

IBM DB2



What is IBM DB2?

IBM® DB2® is the database of choice for enterprise-wide solutions. Optimized to deliver industry-leading performance while lowering costs, IBM DB2 offers extreme performance, flexibility, scalability and reliability for any size organization.

Learn how customers are transforming their data center with DB2.



Exceptional availability



Massive scalability



Deployment flexibility



Lower total cost of ownership

- Den seneste udgave af DB2, DB2 11.1, kører på Linux, UNIX, Windows, IBM iSeries og mainframes.

Informix

IBM Software > Information Management > Information Integration & Governance > Informix >

IBM Informix

Simply Powerful

Overview | Features and benefits | Products | Solutions | Downloads and resources

An Intelligent database for Internet of Things

The powerful new generation of technology revolving around the proliferation of connected devices known as the "Internet of Things" is transforming the way we live and do business. Harnessing data from billions of devices depends on the ability to store, access and query all data together, seamlessly, for speed-of-thought insight and local decision making. IBM® Informix® is forging new frontiers with its embeddability and unique ability to seamlessly integrate SQL, NoSQL/JSON, timeseries and spatial data, with a rich set of APIs, including REST, that enhance development simplicity, flexibility and time to market.

[Watch the video \(00:00:30\)](#)

Highlights

Embedded and optimized database that brings intelligence to edge devices for local decision making and seamless integration with cloud and backend analytics.

- **Real-time Analytics:** Informix is a single platform that can power OLTP and OLAP workloads and successfully meet service-level agreements (SLAs) for each
- **Easy to Use:** Informix runs virtually unattended with self-configuring, self-managing and self-healing capabilities
- **Fast, Always-on Transactions:** Provides one of the industry's widest sets of options
- **Best-of-breed embeddability:** Provides a proven embedded data management platform for ISVs and OEMs to deliver

Contact IBM

Considering a purchase?

[Email IBM](#)

[Request a quote](#)

Or call us at: 1-877-426-3774
Priority code: Analytics Platform

Comparing IBM Informix and Oracle Database

See why IBM Informix is the clear choice over Oracle Database for High Availability and Data Replication
→ [Get the analyst report](#)

IBM Informix Warehouse Accelerator

Extreme warehouse speed for fast business decisions
→ [Get the ebook](#)

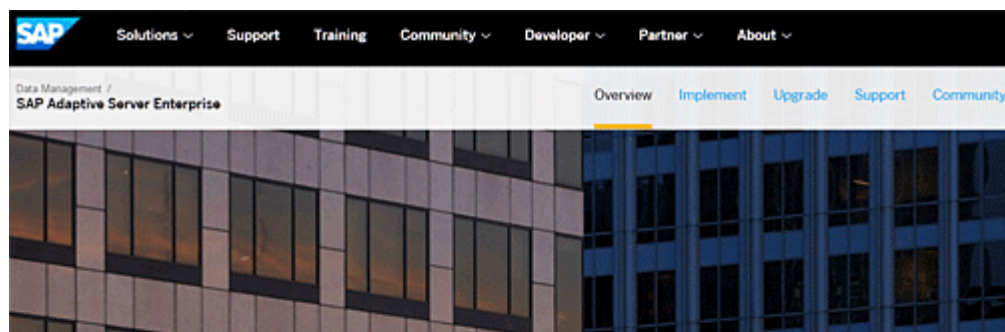
Informix library

Find solutions, industry reports, white papers, videos and more
→ [Visit the library](#)

Related links

- En anden IBM-produkt løsning, Informix har en række versioner, der starter med Workgroup og Express Editions på arbejdsplan og opgraderer til Enterprise Edition, Enterprise Hypervisor Edition og endelig Informix Advanced Workgroup og Enterprise Editions med Informix Warehouse Accelerator (IWA).
- Informix fremhæver sine database-løsninger som "intelligente databaser" til styring af internet af ting (IoT) data, med evnen til problemfrit at integrere SQL, NoSQL / JSON, timeseries og spatial data.

SAP Sybase ASE



Fast-track transaction processing with the SAP ASE relational database server

Drive faster, more reliable online transaction processing (OLTP) – for less – with SAP Adaptive Server Enterprise (formerly SAP Sybase ASE). Designed to meet the demands of the digital economy, this high-performance SQL database server uses a relational model to power transaction-based applications – on premise or in the cloud.

Why SAP

Product Capabilities

System Requirements

Licensing & Purchasing

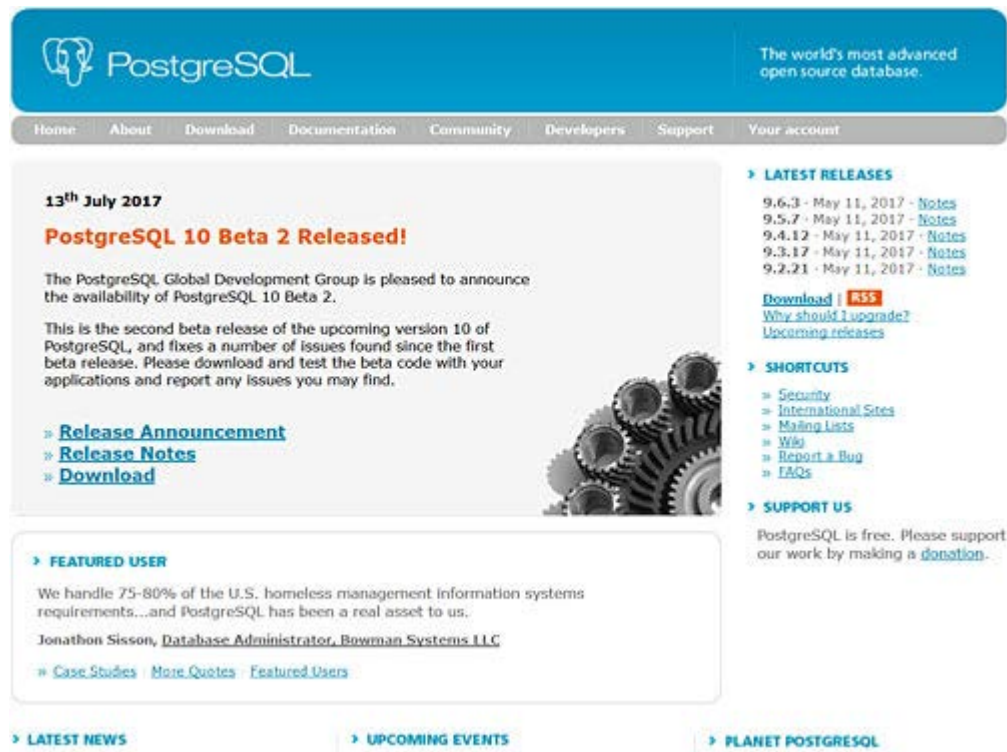
Why SAP ASE?

Because our relational database server modernizes and dramatically speeds up transaction-intensive applications. Meet new demands for performance, reliability, and efficiency – whether you work in financial services, telecommunications, retail,



- Sybase er stadig en stor spiller på enterprise markedet efter 25 års succes og forbedringer af Adaptive Server Enterprise-produktet.
- Selvom markedsandelen faldt i nogle år, har den set en stigning i næste generations transaktions-behandling efter at blive erhvervet af Sybase i 2010 og omdøbt til SAP Sybase Adaptive Server Enterprise (ASE).
- Sybase har også kastet en betydelig vægt bag den mobile virksomhed ved at levere partner løsninger til mobilenhedsmarkedet.

PostgreSQL



- PostgreSQL, eller blot Postgres, er et open-source objekt-relations-baseret database-håndterings-system (ORDBMS), der gemmer sig steder som online spilapplikationer, datacenter-automation og domæne-registre.
- PostgreSQL nyder også nogle højt profilerede opgaver på Skype og Yahoo

MySQL



- Solgt til Sun Microsystems i 2008, MySQL er siden 2009 blevet en del af Oracle-imperiet efter Suns overtagelse af Oracle.
- MySQL driver kommercielle hjemmesider i hundredtusindvis og den tjener også som backend for mange interne virksomheds-applikationer.
- I dag er MySQL stadig en meget populær løsning til brug i webapplikationer, og den fortsætter med at fungere som en central komponent i LAMP open source-applikations-softwarepakken sammen med Linux, Apache og PHP (eller Python eller Perl).
- MySQLs støtte fra brugere og udviklere er eroderet i løbet af årene efter Oracles overtagelse.
- MySQLs tilbagegang har bidraget til at styrke andre open source-databasemuligheder og forks af MySQL som den fuldt open source MariaDB, der ikke indeholder closed source modsat nyere versioner af MySQL Enterprise Edition som Percona og det cloud-optimerede Drizzle databasesystem.

MariaDB Enterprise



MariaDB Server

The world's fastest growing open source database

MariaDB Server is a general purpose database engineered with an extensible architecture to support a broad set of use cases via pluggable storage engines – using different storage engines to support different use cases.

MariaDB Server is one of the most popular open source databases in the world. It is available in Debian and Ubuntu, and it is now the default database on Arch Linux, Manjaro, openSUSE, Red Hat Enterprise Linux/CentOS/Fedora and SUSE Linux Enterprise. And as one of the most broadly adopted and widely deployed in the world, MariaDB Server receives contributions from companies like Alibaba, Facebook, and Google – benefiting the entire community.

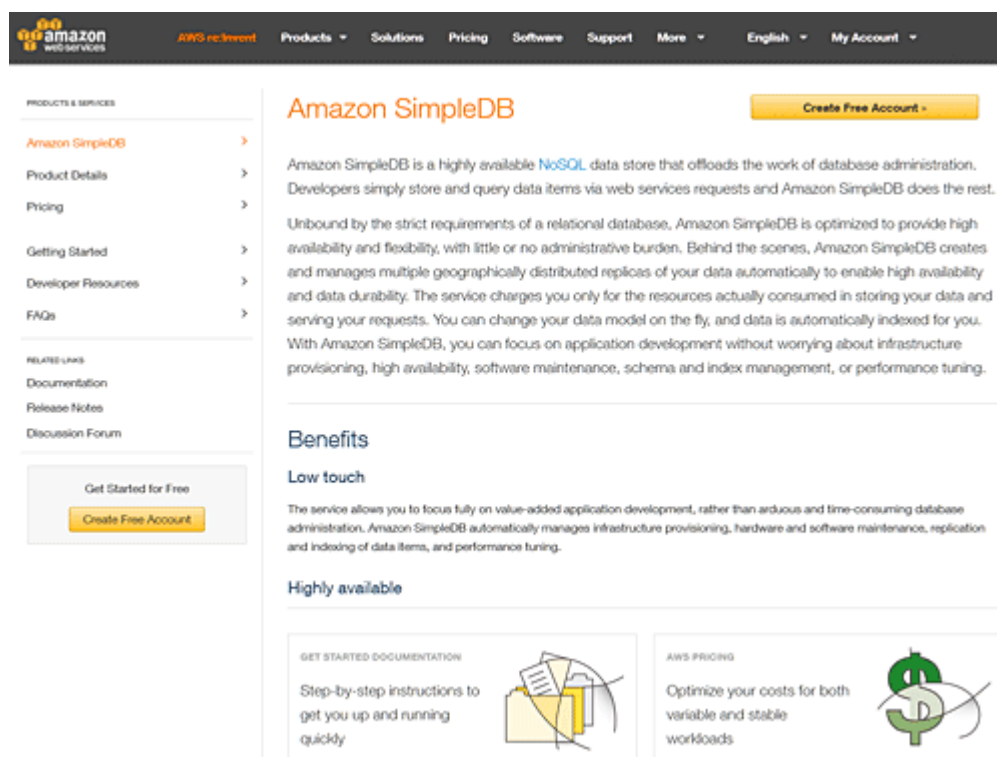
MariaDB Extensible Architecture



The diagram illustrates the MariaDB Extensible Architecture. It shows a stack of components: Applications at the top, followed by Connectors, then the MariaDB Server (which includes MariaDB Multi-Node Cluster and MariaDB Module). The server is connected to the Operating System (OS) and the File System (FS). The architecture is designed to be extensible, allowing for different storage engines and configurations.

- MariaDB Enterprise er et fuldt open source databasesystem med alle kode udgivet under GPL, LGPL eller BSD.
- MariaDB stammer fra 2009 som en community-driven fork af MySQL RDBMS og ledes af de oprindelige udviklere af MySQL, der skabte forken efter bekymringer over Oracles erhvervelse af MySQL.
- MariaDB har set sin popularitet eksplodere for nylig på bekostning af MySQL, især i sin støtte af populære Linux distributioner.
- I 2013 alene gik Red Hat Enterprise Linux (RHEL) MySQL til MariaDB. Fedora valgte MariaDB over MySQL i sin Fedora 19-udgave, og både openSUSE og Slackware Linux lavede lignende skift til MariaDB over MySQL.
- Wikipedia valgte også MariaDB over MySQL som backend-database i 2013.
- En anden vigtig faktor ved at flytte til MariaDB til fordel for MySQL er dets forbedrede forespørgselsoptimering og andre præstationsrelaterede forbedringer, som giver databasesystemet en mærkbar kant i den samlede præstation sammenlignet med MySQL.

Amazon SimpleDB



The screenshot shows the Amazon SimpleDB product page. The top navigation bar includes the Amazon logo, 'AWS re:invent', and links for Products, Solutions, Pricing, Software, Support, and More. A 'Create Free Account' button is visible. The left sidebar lists 'PRODUCTS & SERVICES' with 'Amazon SimpleDB' selected, and 'RELATED LINKS' including Documentation, Release Notes, and Discussion Forum. The main content area features the 'Amazon SimpleDB' title, a description of it as a highly available NoSQL data store, and a 'Benefits' section with 'Low touch' and 'Highly available' subsections. At the bottom, there are two boxes: 'GET STARTED DOCUMENTATION' with a folder icon and 'AWS PRICING' with a dollar sign icon.

Amazon SimpleDB [Create Free Account](#)

Amazon SimpleDB is a highly available NoSQL data store that offloads the work of database administration. Developers simply store and query data items via web services requests and Amazon SimpleDB does the rest.

Unbound by the strict requirements of a relational database, Amazon SimpleDB is optimized to provide high availability and flexibility, with little or no administrative burden. Behind the scenes, Amazon SimpleDB creates and manages multiple geographically distributed replicas of your data automatically to enable high availability and data durability. The service charges you only for the resources actually consumed in storing your data and serving your requests. You can change your data model on the fly, and data is automatically indexed for you. With Amazon SimpleDB, you can focus on application development without worrying about infrastructure provisioning, high availability, software maintenance, schema and index management, or performance tuning.

Benefits

Low touch

The service allows you to focus fully on value-added application development, rather than arduous and time-consuming database administration. Amazon SimpleDB automatically manages infrastructure provisioning, hardware and software maintenance, replication and indexing of data items, and performance tuning.

Highly available

GET STARTED DOCUMENTATION

Step-by-step instructions to get you up and running quickly



AWS PRICING

Optimize your costs for both variable and stable workloads



- Databaser og Amazon kan i første omgang synes verdener fra hinanden, men det er de ikke, især i vores stadig mere cloud-centrerede verden.
 - Amazons SimpleDB (Simple Database Service) tilbyder virksomheder et simpelt, fleksibelt og billigt alternativ til traditionelle databasesystemer.
- SimpleDB gør det muligt for brugere at gemme og søge dataelementer via webserviceforespørgsler, og det kan prale af skalerbarhed, hastighed, minimal vedligeholdelse og Amazon services integration.
- Som en del af Amazons EC2-tilbud kan du komme i gang med SimpleDB gratis.

Den relationelle model

En opsummering

Hvad er den relationelle model

- Den relationelle datamodel er den primære datamodel, som bruges over hele verden til datalagring og -behandling.
 - Denne model er enkel, og den har alle de egenskaber og funktioner, der kræves til at behandle data med lagringseffektivitet.
-
- Relationelle databasesystemer forventes at være udstyret med et forespørgselssprog, som kan hjælpe sine brugere med at forespørge om databasernes forekomster. Der findes to slags forespørgselssprog - relationel algebra og relationelle kalkulus (beregninger).
 - SQL bytter videre oven på disse sprog-tanker, og derfor bruger vi ikke tid derpå.

SQL databasens historie

- Den moderne database har rødder tilbage i 1960erne takket være bl.a. IBM.
- I 1970 udgav Edgar “Ted” Codd, der også var ansat ved IBM, en artikel der beskriver en relationel. Han lagde senere navn til en af normal-formerne (*Boyce–Codd normal formen*).
- IBM var dog langsomme til at implementere den relationelle database, og lavede således det ikke relationelle sprog SEQUEL, der var så godt at det allerede inden udgivelse blev kopieret og brugt i Oracle Database af Larry Ellison og navnet blev skiftet til SQL.
- Det blev en vedtaget ANSI standard i 1986 og ISO i 1987, og selv om der i dag findes mange mere eller mindre proprietære implementeringer er det grundlæggende sprog universelt mellem mange database systemer.

Koncepterne bag den relationelle model

- **Tables** (tabeller) – I relationel datamodel gemmes relationer i tabeller. Dette format lagrer forholdet mellem entities (enheder). En tabel har rækker og kolonner, hvor rækker repræsenterer poster og kolonner repræsenterer attributterne.
- **Tuple** (tupel) – En enkelt række i en tabel, som indeholder en enkelt post for et forhold, hedder en tuple.
- **Relation instance** (relations instans) – Et finite (begrænset) sæt tupler i relationsdatabasesystemet repræsenterer relative forekomster. Relations instanser har ikke duplikat tuples.

Koncepterne bag den relationelle model

- **Relation schema** (**relations** skema) – Et relationsschema beskriver forholdets navn (tabelnavn), attributter og deres navne.
- **Relation key** (**relations nøgle**) – Hver række har en eller flere attributter, kendt som relations nøgle, som kan identificere rækken i relationen (tabellen) entydigt.
- **Attribute domain** (**attribute domæne**) – Hver attribut har noget foruddefineret value scope (værdiomfang), kendt som attributdomenet.

Constraints (begrænsninger)

- Hvert forhold har nogle betingelser, der skal opretholdes for at danne et gyldigt forhold. Disse betingelser kaldes **Relational Integrity Constraints** (**Relationelle Integritets Begrænsninger**).
- Der er tre primære begrænsninger:
 - Key constraints (nøgle begrænsninger)
 - Domain constraints (domæne begrænsninger)
 - Referential integrity constraints (Reference integritet begrænsninger)

Key constraints (nøgle begrænsninger)

- Der skal være mindst en minimal delmængde af attributter i forholdet, som kan identificere en tuple entydigt.
- Denne minimal delmængde af attributter kaldes en nøgle for den relation.
- Hvis der er mere end en sådan minimal delmængde, kaldes disse kandidatnøgler.
- Nøglebegrænsninger gennemtvinger at
 - i et forhold til en nøgleattribut, kan ingen to tupler have identiske værdier for nøgleattributter.
 - en nøgleattribut kan ikke have NULL-værdier.
- Nøglebegrænsninger kaldes også Entity Constraints (entitets begrænsninger).

Domain constraints (domæne begrænsninger)

- Attributter har specifikke værdier i real-world scenarier.
 - For eksempel kan alder kun være et positivt heltal.
- De samme begrænsninger er blevet forsøgt at benyttet på attributterne af en relation.
- Hver egenskab er forpligtet til at have et specifikt udvalg af værdier.
 - For eksempel kan alderen ikke være mindre end nul, og telefonnumre kan ikke indeholde et tegn uden for 0-9.

Referential integrity constraints (Reference integritet begrænsninger)

- Referential integritet begrænsninger fungerer ud fra begrebet foreign keys (fremmed nøgler).
- En fremmed nøgle er en nøgleattribut til et forhold, der kan henvise til et andet forhold.
- Referential integritets begrænsning angiver, at hvis en relation refererer til en nøgle attribut for et andet eller lignende forhold, så skal nøgleelementet eksistere.

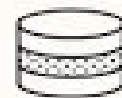
Normalformer uddybet

Resume og flere former

Normalformens historie

- Edgar F. Codd, der også “opfandt” relationsmodellen, introducerede Den Første Normal Form i 1970 og anden og tredje form i 1971.
- I 1977 føjede Ronald Fagin en Fjerde og en Femte Normal Form til.
- Den nyeste tilføjelse af den Sjette Normal Form fra 2002 af C.J. Date, Hugh Darwen og Nikos Lorentzos
- Der er andre former ud over disse, f.eks. den tidligere nævnte Boyce Codd Normal

Opsummerende video af konceptet



Database

NORMALIZATION

Problemer uden normalisering

- Uden Normalization er det svært at håndtere og opdatere database uden at risikere data tab. Insert, Update og Delete anomalier er meget hyppige hvis en database ikke er normaliseret.
- Lad os kigge på **Student** tabellen for at illustrere dette:

Student	Age	Subject
Adam	15	Biology, Maths
Alex	14	Maths
Stuart	17	Maths

Problemer uden normalisering

- De problemer vi møder er:
 - **Updation Anomaly:** For at opdatere adressen på en elev, der er opført to gange eller mere i en tabel, skal vi opdatere **S_Address** kolonnen i alle rækker, ellers bliver data inkonsekvent.
 - **Insertion Anomaly:** Antag for en ny optagelse, hvor vi har en Student ID (S_id), navn og adresse på den studerende, men at den studerende ikke har valgt noget fag endnu, så skal vi indsætte **NULL** der, hvilket leder til Insertion Anomaly.
 - **Deletion Anomaly:** Hvis (S_id) 401 kun har et fag og midlertidigt dropper det, når vi så sletter rækken vil hele studenter optegnelsen ryge dermed.

Tabellen i 1ste normal form

Student	Age	Subject
Adam	15	Biology
Adam	15	Maths
Alex	14	Maths
Stuart	17	Maths

Tabellen i 1ste normal form



1st Normal Form (1NF)

DBMS NORMALIZATION

Problemer uden 2nden normal form

- I eksempelet for Første Normal Form er der to rækker til Adam, der omfatter flere emner, som han har valgt.
- Selvom dette er søgbart, og følger Første normale form, er det en ineffektiv udnyttelse af pladsen.
- Også i ovenstående tabel i første normale form, mens kandidatnøglen er {Student, Emne}, afhænger Age af Student kun af Student-kolonnen, hvilket er forkert i anden normal form.
- For at opnå anden normal form ville det være nyttigt at opdele emnerne i en selvstændig tabel og sammenstille dem med de studerendes navne som fremmednøgler.

Tabellen i 2nden normal form

- I Student tabellen er kandidat nøglen **Student** kolonnen, da alle andre kolonner, dvs. **Age**, er afhængig af den.

Student	Age
Adam	15
Alex	14
Stuart	17

Tabellen i 2nden normal form

- I Subject tabellen er kandidat nøglen {**Student**, **Subject**} kolonnen.

Student	Subject
Adam	Biology
Adam	Maths
Alex	Maths
Stuart	Maths

- Nu er begge tabeller i Anden Normal Form og vil aldrig lide af Update anomalier.
 - Der er dog et par komplekse tilfælde, hvor tabeller i anden normal form kan lide af opdateringsanomalier, for at håndtere disse scenarier, er den tredje normale form der.

Tredje normal form

- Tredje normalform gør at alle ikke-primære attributter i tabellen skal være afhængige af primærnøglen.
 - En anden værdi må ikke være direkte afhængig af en anden værdi, kun af primær nøglen.
- Lad os se på denne **Student_Detail** tabel

Student_id	Student_name	DOB	Street	city	State	Zip
------------	--------------	-----	--------	------	-------	-----

Tabellen i 3die normal form

- I tabellen på sidste slide er Student_id primærnøgle, men gade, by og stat afhænger af Zip.
- Afhængigheden mellem zip og andre felter kaldes transitiv afhængighed.
- Derfor skal vi anvende 3NF til at flytte gaden, byen og staten til en ny tabel, med Zip som primærnøgle.
- Ny **Student_Detail** tabel

Student_id	Student_name	DOB	Zip
------------	--------------	-----	-----

- **Adresse** tabel

Zip	Street	city	state
-----	--------	------	-------

Opsummering af de første tre (de essentielle)

- Den **første normalform** definere at der må ikke være forskellig data i samme felt, hvilket betyder at informationen deles i tabeller efter relation.
- Ligeledes fjernes gentagne grupper fra individuelle tabeller, hvor hvert sæt af relateret data har en primær-nøgle.
- I **anden normalform** fjernes den redundante data, således sættes værdisæt, der kan bruges til flere poster, i særskilte tabeller.
- I **tredje normalform** bliver det, der ikke beskriver primærnøglen fjernet eller flyttet til andre tabeller.

Opsummering af de første tre (de essentielle)

- Essensen af de første tre normalformer er altså at dataen i tabellen skal beskrive nøglen.
- I web-udvikling plejer man normalt kun at gå op i at opfylde de første tre normal-former, da det er de mest essentielle, og de følgende former leder til strukturer der ikke nødvendigvis er optimale til denne brug. Dette ledte til den klassiske huskeregel inspireret af den amerikanske ed ved domstolene: “The data should depend on the key (1NF), the whole key (2NF) and nothing but the key (3NF) (so help med Codd)”.

Boyce and Codd Normal Form (BCNF)

- BCNF er en smule stærkere udgave af 3. normalform, derfor kaldes den også 3.5NF
 - Den er lavet til at dække nogle anomalier som 3NF ikke dækker.
- Kun i sjældne tilfælde opfylder en 3NF-tabel ikke kravene i BCNF.
- En 3NF-tabel, der ikke har flere overlappende kandidatnøgler, garanteres at være i BCNF.
- Afhængigt af dens funktionelle afhængigheder er, kan en 3NF-tabel med to eller flere overlappende kandidatnøgler være eller ikke være i BCNF.

Boyce and Codd Normal Form (BCNF)

- Et eksempel på en 3NF tabel der ikke opfylder BCNF

Today's Court Bookings

Court	Start Time	End Time	Rate Type
1	09:30	10:30	SAVER
1	11:00	12:00	SAVER
1	14:00	15:30	STANDARD
2	10:00	11:30	PREMIUM-B
2	11:30	13:30	PREMIUM-B
2	15:00	16:30	PREMIUM-A

- I Today's Court Bookings-tabel er der ingen ikke-primære attributter: det vil sige alle attributter tilhører en kandidatnøgle. Derfor opfylder tabellen både 2NF og 3NF.
- Tabellen overholder ikke BCNF. Dette skyldes afhængighedsfrekvenstype → Court, hvori den bestemmende attributsfrekvenstype, som Court afhænger af, hverken er en kandidatnøgle eller et supersæt af en kandidatnøgle.
- Afhængighedsfrekvenstype → Court respekteres, da en Rate Type kun skal kunne gælde for en enkelt Court.

Boyce and Codd Normal Form (BCNF)

- Tabellen delt i to så den opfylder BCNF

Rate Types			Today's Bookings			
Rate Type	Court	Member Flag	Member Flag	Court	Start Time	End Time
SAVER	1	Yes	Yes	1	09:30	10:30
STANDARD	1	No	Yes	1	11:00	12:00
PREMIUM-A	2	Yes	No	1	14:00	15:30
PREMIUM-B	2	No	No	2	10:00	11:30
			No	2	11:30	13:30
			Yes	2	15:00	16:30

- Kandidatnøglerne for tabellen Rate Types er {Rate Type} og {Court, Member Flag};
- Kandidatnøglerne til Today's Bookings-tabel er {Court, Start Time} og {Court, End Time}.
- Begge tabeller er i BCNF.
- Når {Rate Type} er en nøgle i tabellen Rate Types er det umuligt at have en Rate Type tilknyttet to forskellige Courts, så ved at bruge {Rate Type} som en nøgle i tabellen Rate Types er uregelmæssigheden, der påvirker originaltabellen, blevet elimineret.

Fjerde normal form

- Her må en tabel ikke have flere en-til-mange og mange-til-en relationer, der ikke direkte hænger sammen og en værdi må ikke have flere afhængigheder.

Femte normal form

- Alle ikke-trivielle joins skal relatere til primær-nøglen, hvilket betyder at det nogle gange sparer resurser at dele ellers relateret data op.

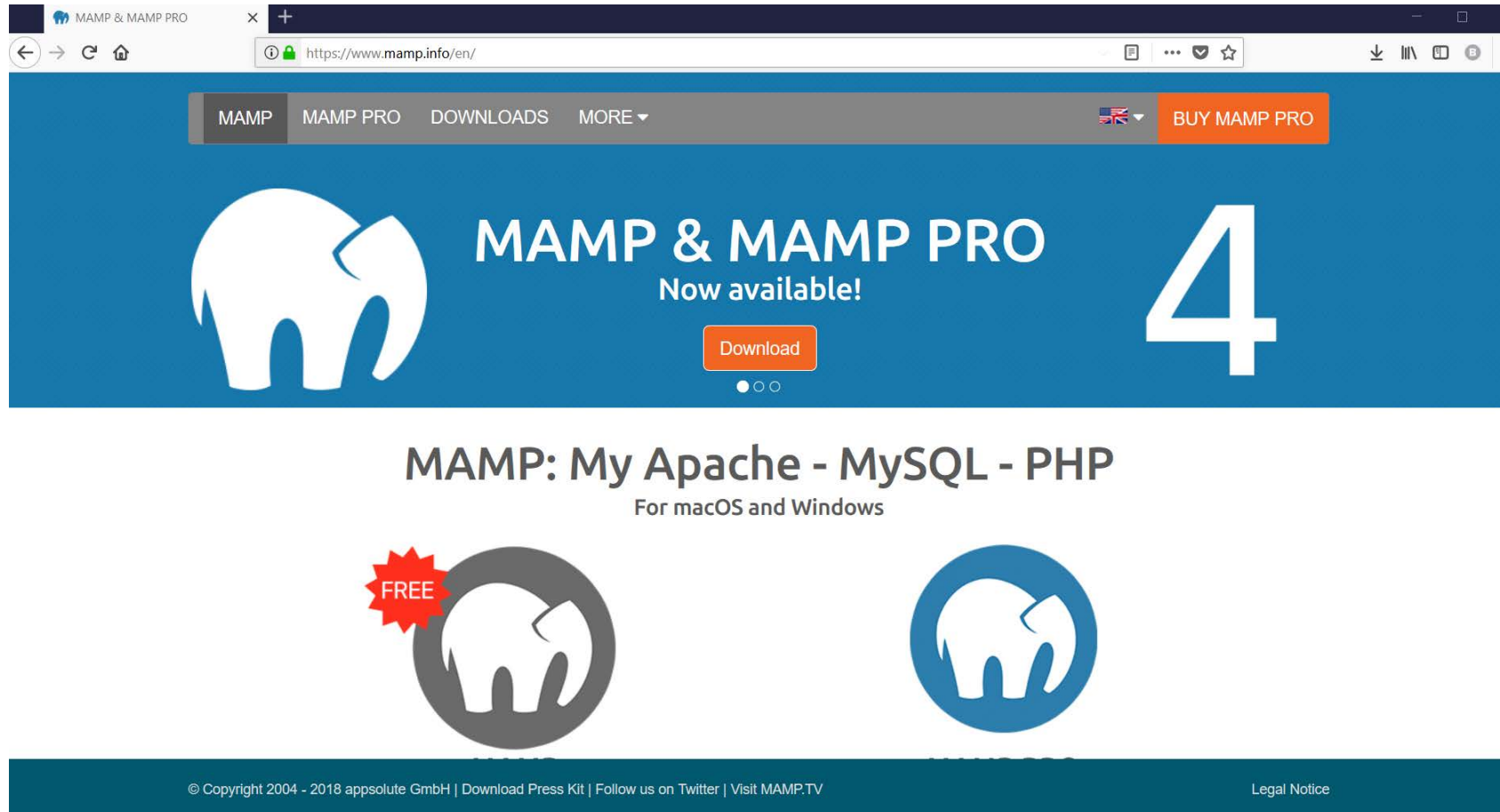
Sjette normal form

- Den sjette normalform er ret radikal da der ikke må være nogen trivielle join afhængigheder overhovedet.
- Den bruges i store datavarehuse og resulterer i rigtig mange tabeller, hvoraf de fleste droppes undervejs.

MAMP

Lokal web (PHP) og MySQL database-server pakke installeres

Gå til www.mamp.info og hent softwaren



The screenshot shows the homepage of the MAMP website. At the top, there is a navigation bar with links for MAMP, MAMP PRO, DOWNLOADS, and MORE. A language selector (UK flag) and a 'BUY MAMP PRO' button are also present. The main banner features a large white elephant logo on a blue background, the text 'MAMP & MAMP PRO' in white, 'Now available!' in smaller white text, a 'Download' button, and a large white number '4'. Below the banner, the text 'MAMP: My Apache - MySQL - PHP' is displayed, followed by 'For macOS and Windows'. Two circular logos are shown: a grey one with a red starburst saying 'FREE' and a blue one. The footer contains copyright information: '© Copyright 2004 - 2018 appbsolute GmbH | Download Press Kit | Follow us on Twitter | Visit MAMP.TV' and a 'Legal Notice' link.

MAMP & MAMP PRO
Now available!
Download

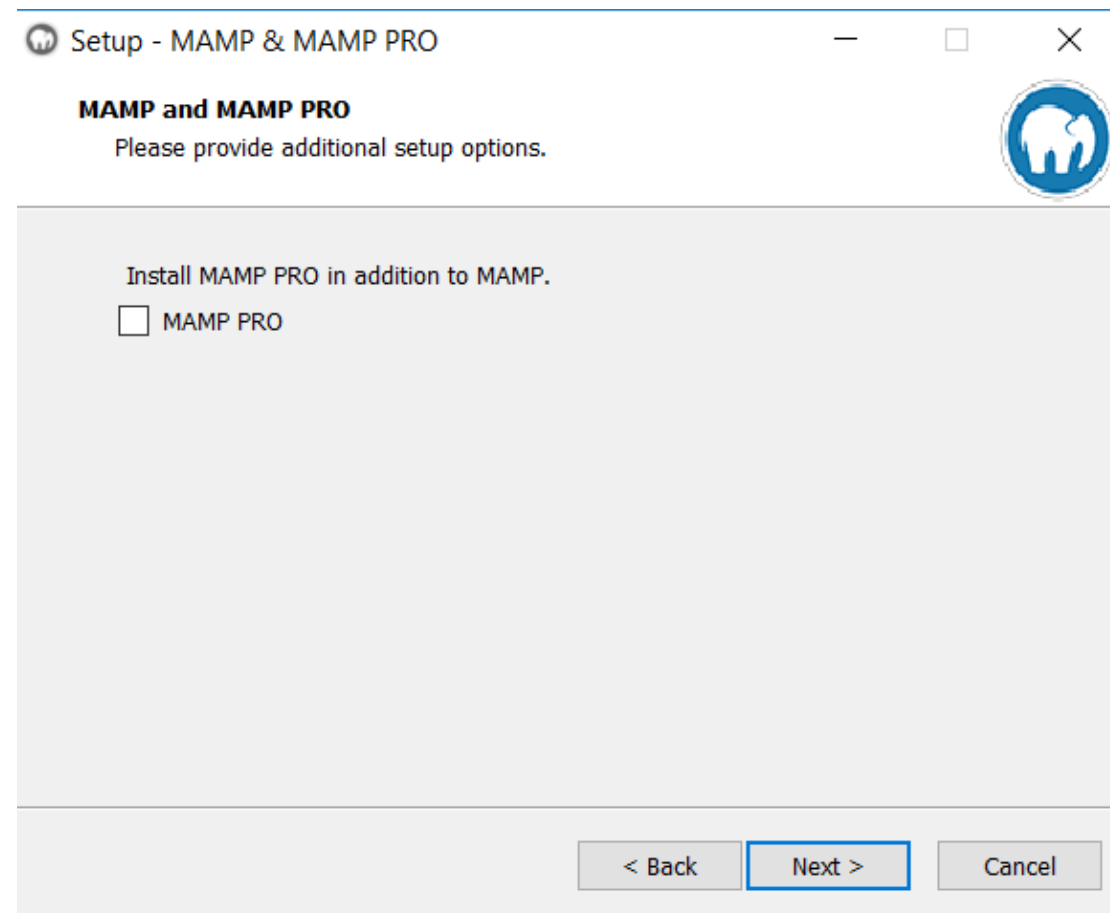
4

MAMP: My Apache - MySQL - PHP
For macOS and Windows

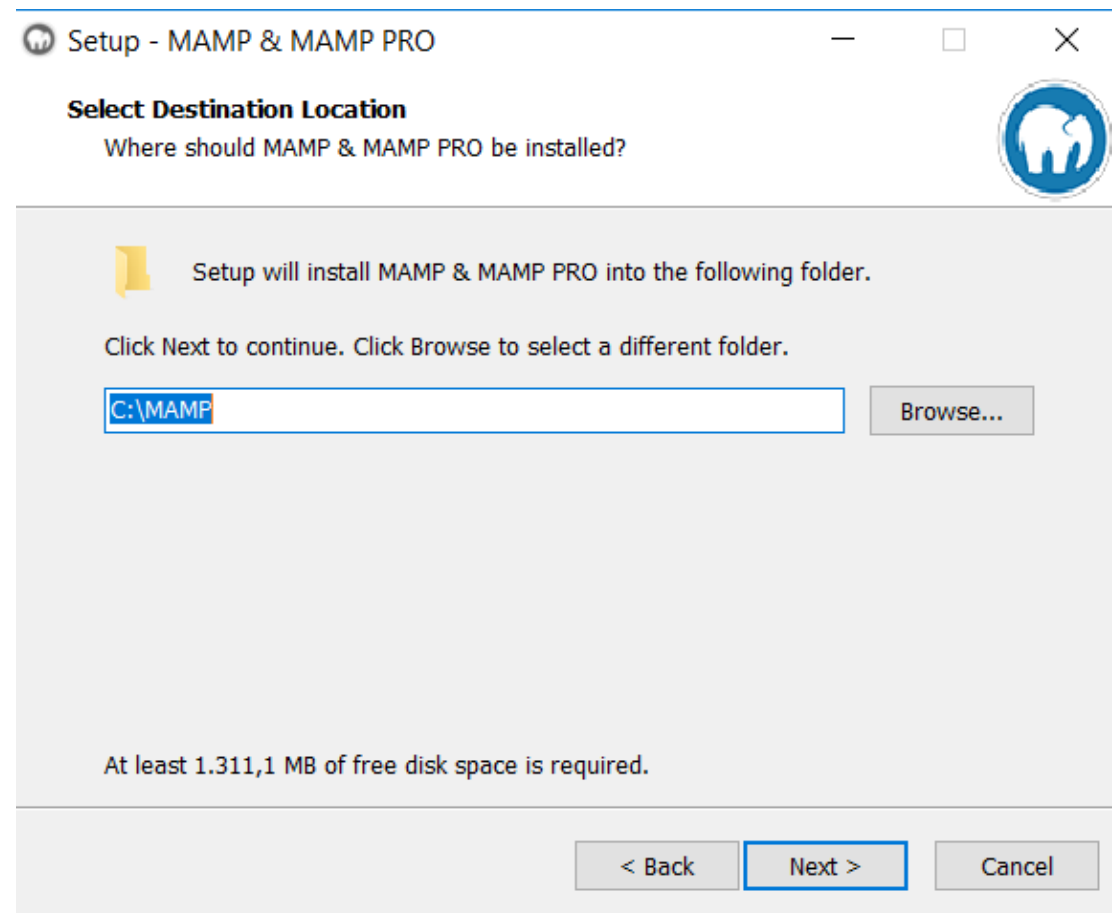
FREE

© Copyright 2004 - 2018 appbsolute GmbH | Download Press Kit | Follow us on Twitter | Visit MAMP.TV
Legal Notice

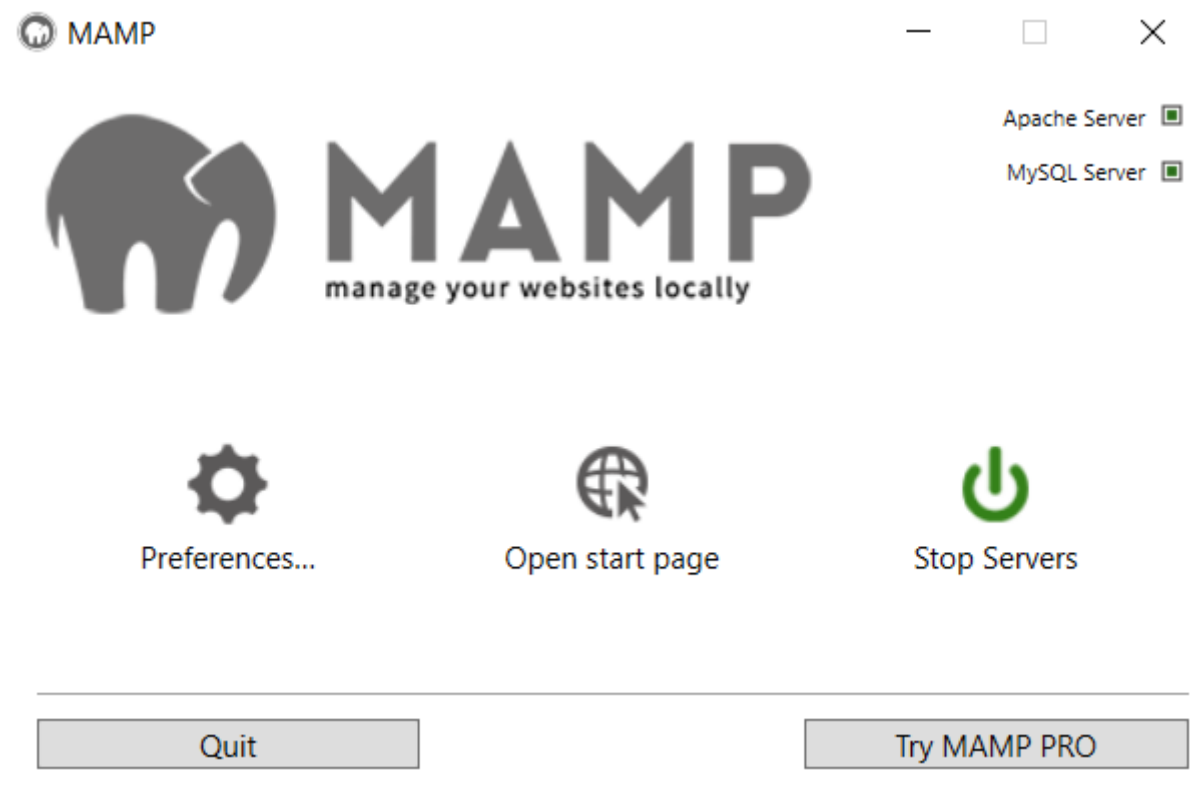
I har ikke behov for at installere MAMP PRO med



Hvor I vælger at lægge web dokumenter er ikke så vigtigt da vi primært vil bruge database delen



Når I starter MAMP ser I at der kører en MySQL Server



Når I er på MAMPs start side vil I under Tools finde phpMyAdmin, herfra styres MySQL databasen

MAMP

localhost/MAMP/

Start My Website phpInfo Tools FAQ MAMP Website Buy MAMP PRO

phpMyAdmin
SQLite Manager (needs PHP 5.2.x or 5.3.x)
phpLiteAdmin
APC (not loaded)
eAccelerator (not loaded)
XCache (not loaded)
OPcache (not loaded)

PHP
phpinfo shows the current configuration of PHP.

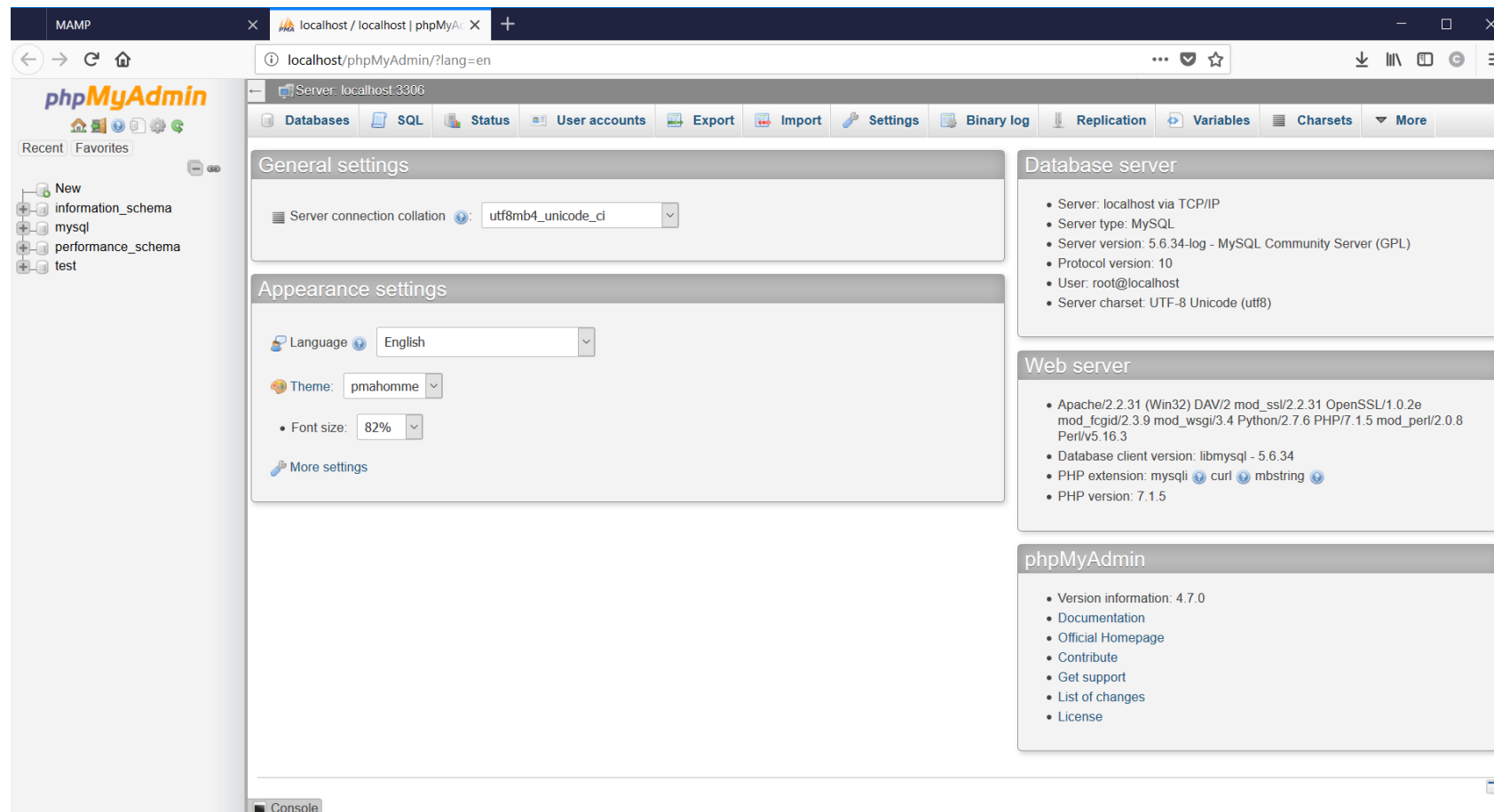
MySQL
MySQL can be administered with [phpMyAdmin](#).
To connect to the MySQL server from your own scripts use the following connection parameters:

Host	localhost
Port	3306
User	root
Password	root
Socket	/Applications/MAMP/tmp/mysql/mysql.sock

MAMP Version
3.3.1 → Your version is up-to-date.

News
MAMP & MAMP PRO 4.1.1 out now
February, 2017 – **Version 4.1.1 is out now** - comes with **German localization**
- Added german localisation
- Initial TouchBar support on MacBook Pro 2016
- Automatically deleting a database when deleting a host fixed
- Some functions can now only be used if the active servers are using the current configuration
- Solved PHP 7.1 configuration quirks

Sådan ser phpMyAdmin ud



Mere end CRUD

Vi læser, skriver og sletter mere

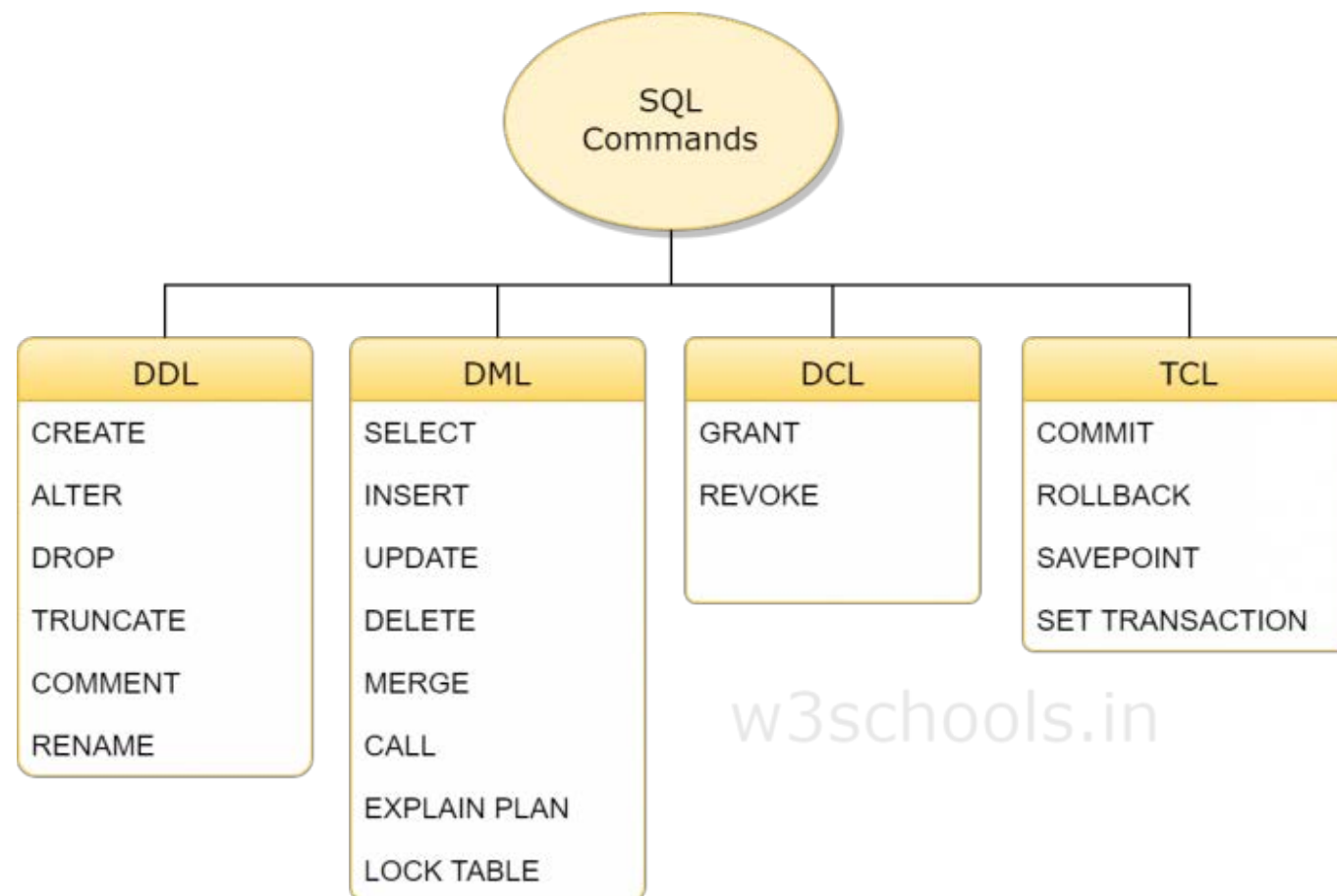
Essensen af SQL

- Klausuler (**clauses**), der er konstituerende bestanddele af erklæringer og forespørgsler – i nogle tilfælde valgfri.
- Udtryk (**expressions**), der kan producere enten skalar værdier, eller tabeller, der består af kolonner og rækker af data.
- Prædikater (**predicates**), der angiver betingelser, der kan evalueres til SQL tre-værdi logik
- (3VL) (sandt / falsk / ukendt) (*three-valued logic*) eller Boolske værdier (boolsk algebra), der anvendes til at begrænse virkningerne af erklæringer og forespørgsler, eller til at ændre programmets flow.
- Forespørgsler (**queries**), der henter data baseret på specifikke kriterier. Dette kan betegnes ssom essensen af SQL.

Essensen af SQL

- Udsagn (**statements**), som kan have en vedvarende effekt på skemaer (schema, samling af database objekter, typisk tabeller) og data, eller som kan kontrollere transaktioner, programmets flow, forbindelser, sessioner, eller diagnostik.
 - SQL-sætninger benytter semikolon (";") terminatoren. Selv om det ikke kræves på alle platforme, det er defineret som en standard del af SQL-grammatik.
- Ubetydelige mellemrum er generelt ignoreret i SQL-sætninger og forespørgsler, hvilket gør det let at formatere SQL kode læsbart.

SQL kommandoer kan deles i fire grupper



w3schools.in

DDL

- DDL er en forkortelse for **Data Definition Language**, der omhandler databaseskemaer og beskrivelser af, hvordan data skal opholde sig i databasen.
 - CREATE – at lave databasens og dens objekter (table, index, views, store procedure, function, og triggers)
 - ALTER – ændrer strukturen på en eksisterende database
 - DROP – sletter objekter fra databasen
 - TRUNCATE – fjerner alle optegnelser fra en table, inklusiv al plads allokeret til optegnelserne
 - COMMENT – føjer kommentarer til data dictionary (ordbogen)
 - RENAME – omdøber et objekt

DML

- DML er kort for **Data Manipulation Language** der har med data manipulation at gøre og indkluderer de mest udbredte SQL statements som SELECT, INSERT, UPDATE, DELETE etc, og bruges til at gemme, modificere, hente, slette og opdatere data i en database.
 - SELECT – henter data fra en database
 - INSERT – indsætter data i en tabel
 - UPDATE – opdaterer eksisterende data i en tabel
 - DELETE – sletter alle optegnelser fra en database tabel
 - MERGE – UPSERT operation (insert eller update)
 - CALL – kald et PL/SQL eller Java subprogram
 - EXPLAIN PLAN – interpretation af data access path (data adgangs stien)
 - LOCK TABLE – concurrency control (samtidigheidskontrol)

DCL

- DCL er kort for **Data Control Language** der indeholder kommandoer som GRANT og primært bekymrer sig om rights (rettigheder), permissions (tilladelser) og andre controller af database systemet.
 - GRANT – tillader brugere adgangs privilegier til databasen
 - REVOKE – trækker brugeres adgangs privilegier givet med GRANT kommandoen tilbage

TCL

- TCL er kort for **Transaction Control Language** der har med transaktioner inde i en database at gøre.
 - COMMIT – udfører en Transaction
 - ROLLBACK – ruller en transaktion tilbage i tilfælde af at der sker en fejl
 - SAVEPOINT – laver et retur punkt for transaktionerne i grupper
 - SET TRANSACTION – specificerer egenskaberne for transaktionen

Dette kursus

- I dette kursus kigger vi dog primært på nogle af elementerne inden for DDL og DML, de resterende samt DCL og TCL hører under **videregående database**
- Lad os se på nogle af SQL kommandoerne og bruge dem i praksis

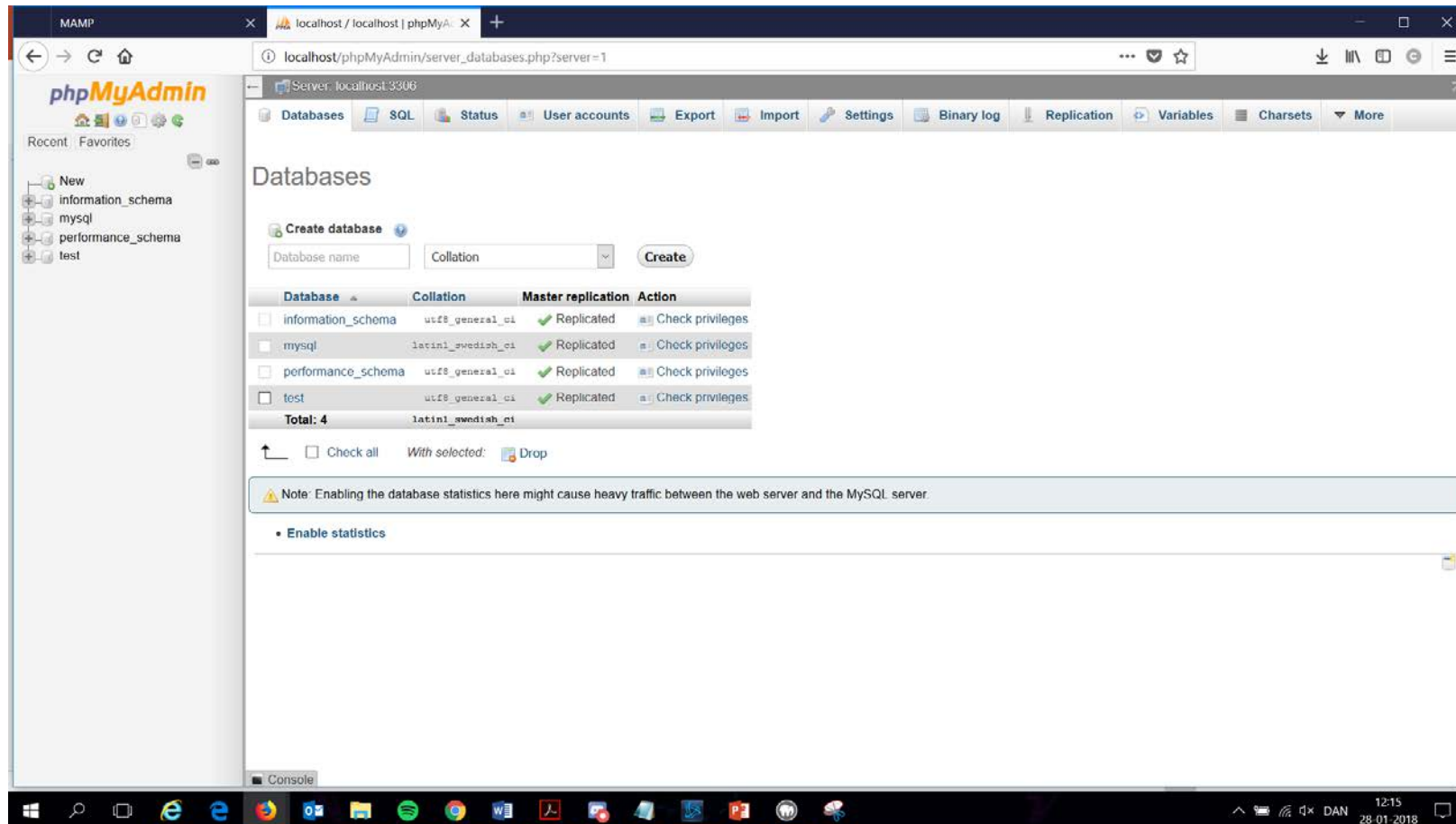
Create

- Man laver en database og tabeller med create

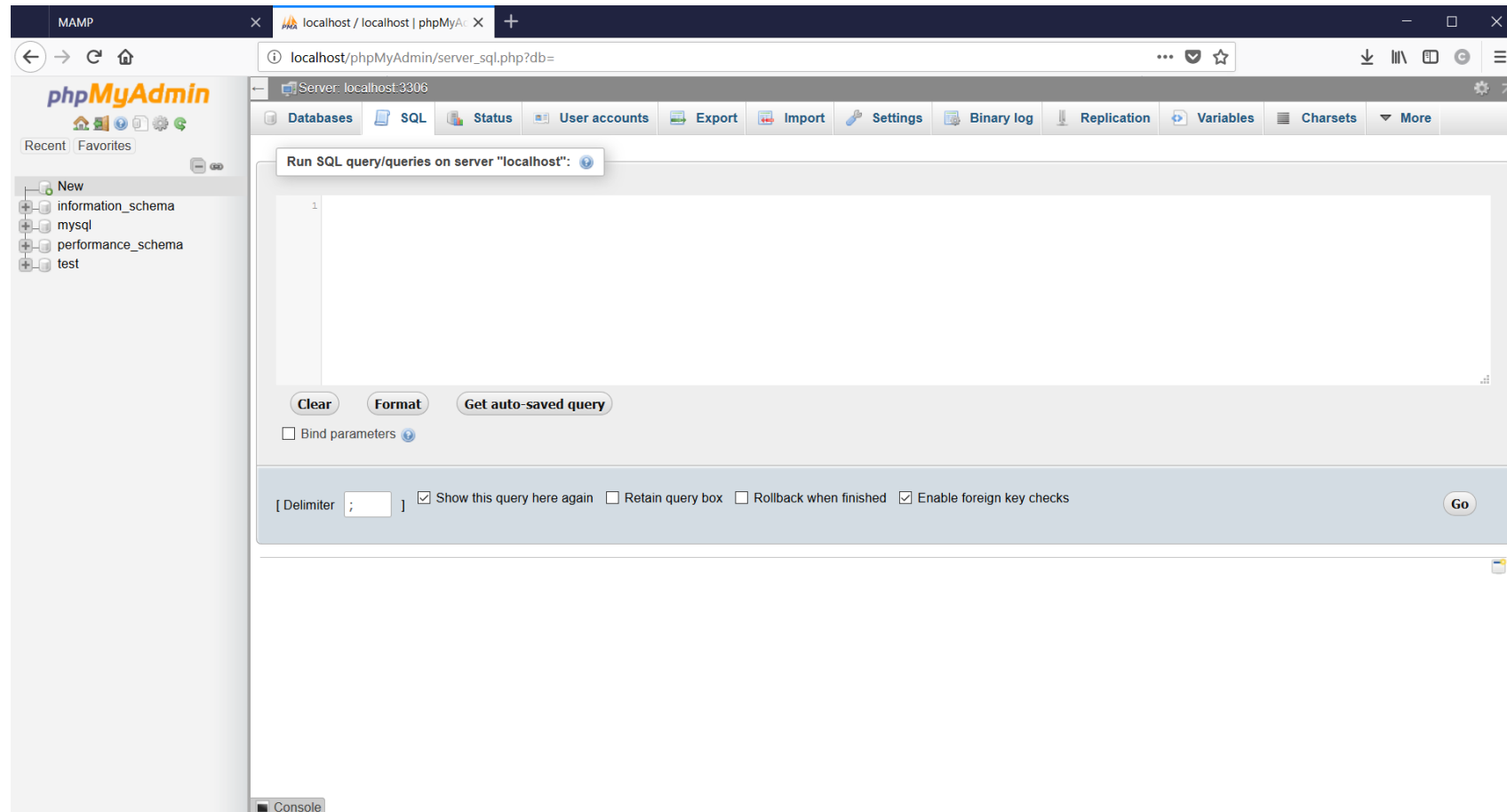
```
CREATE DATABASE [Name];
```

```
CREATE TABLE [Name] ([Column1 name] [Type],  
[Column2 name] [Type]);
```

Vi laver en ny database via phpMyAdmin



Vælg SQL da vi gør det via kode



Lav en ny database og tabel

```
CREATE DATABASE mydb;
```

```
USE mydb;
```

```
CREATE TABLE mytable (cpr BIGINT(10), name  
VARCHAR(50));
```

Vi har nu en database med en tabel i

The screenshot shows the phpMyAdmin web interface in a browser window. The address bar indicates the URL is `localhost/phpMyAdmin/db_structure.php?db=&db=mydb`. The left sidebar shows the database structure with 'mydb' selected, containing a table 'mytable'. The main panel displays the 'Structure' tab for 'mytable'.

Filters

Containing the word:

Table	Action	Rows	Type	Collation	Size	Overhead
☐ mytable	Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	16 K B	-
1 table	Sum	0	InnoDB	latin1_swedish_ci	16 K B	0 B

☐ Check all With selected:

[Print](#) [Data dictionary](#)

[Create table](#)

Name: Number of columns:

Where

- Den primære betingelsesklausul (conditional clause)

WHERE [key] = "[value]"

!=

<

>

<=

>=

OR [key] = "[value]"

AND [key] = "[value]"

Vi sætter først data ind og henter derefter noget af det

Husk at sørge for at være i mydb databasen når I kører SQL kommandoerne!

```
INSERT INTO mytable (cpr,name)
VALUES (2703820101, "Erik Weber-Lauridsen"),
(2403870102, "Britta Manuela Weber-Lauridsen")
```

- Kør derefter:

```
SELECT name FROM mytable
WHERE cpr !=2703820101
```

Limit

- LIMIT-klausulen begrænser mængden af resultater
 - Det vil sige at uanset hvor mange resultater du får vises de [Int] første kun
- God til pagination

```
SELECT * FROM [Table]  
LIMIT [Int]
```

Order By

- Sorter efter en værdi, både stigende eller faldende

`ORDER BY [value] [ASC/DESC]`

Nøgler låser data op

Identifikation og kryds-referencer mellem tabeller

Hvad er nøgler

- En nøgle er et enkelt eller en kombination af flere felter i en tabel.
- Den bruges til at hente eller hente poster / data-rækker fra datatabellen i henhold til betingelsen / kravet.
- Nøgler bruges også til at skabe forhold mellem forskellige databasetabeller eller visninger.

Nøgle typer

- Der er følgende typer nøgler i SQL, som bruges til at hente poster fra tabeller og for at skabe forhold mellem tabeller eller visninger.

1. Super nøgle

En supernøgle er et sæt af en eller flere nøgler, der kan bruges til at identificere en optegnelse unikt i en tabel.

Eksempel: Primær nøgle, Unik nøgle, en Alternate (Alternativ) nøgle er en delmængde af Super nøgler.

2. Kandidat nøgle

En kandidatnøgle er et sæt af et eller flere felter / kolonner, som kan identificere en post unikt i en tabel. Der kan være flere kandidatnøgler i en tabel. Hver kandidatnøgle kan fungere som primærnøgle.

Eksempel: I følgende diagram er ID, RollNo og EnrollNo kandidatnøgler, da alle disse tre felter kan fungere som primærnøgle.

Nøgle typer

3. Primær nøgle

En primærnøgle er et sæt af et eller flere felter / kolonner af en tabel, der unikt identificerer en post i database tabel. Den kan ikke acceptere null, duplikatværdier. Kun en kandidatnøgle kan være primærnøgle.

4. Alternate (Alternativ) nøgle

En alternativ nøgle er en nøgle, som kan fungere som en primær nøgle. Dybest set er det en kandidatnøgle, der i øjeblikket ikke er primærnøgle.

Eksempel: I følgende diagram bliver RollNo og EnrollNo alternativ nøgler, når vi definerer ID som primærnøgle.

Nøgle typer

5. Composite/Compound (komposit) nøgle

En komposit nøgle er en kombination af mere end et felt / kolonner i en tabel. Det kan være en kandidatnøgle, primærnøgle.

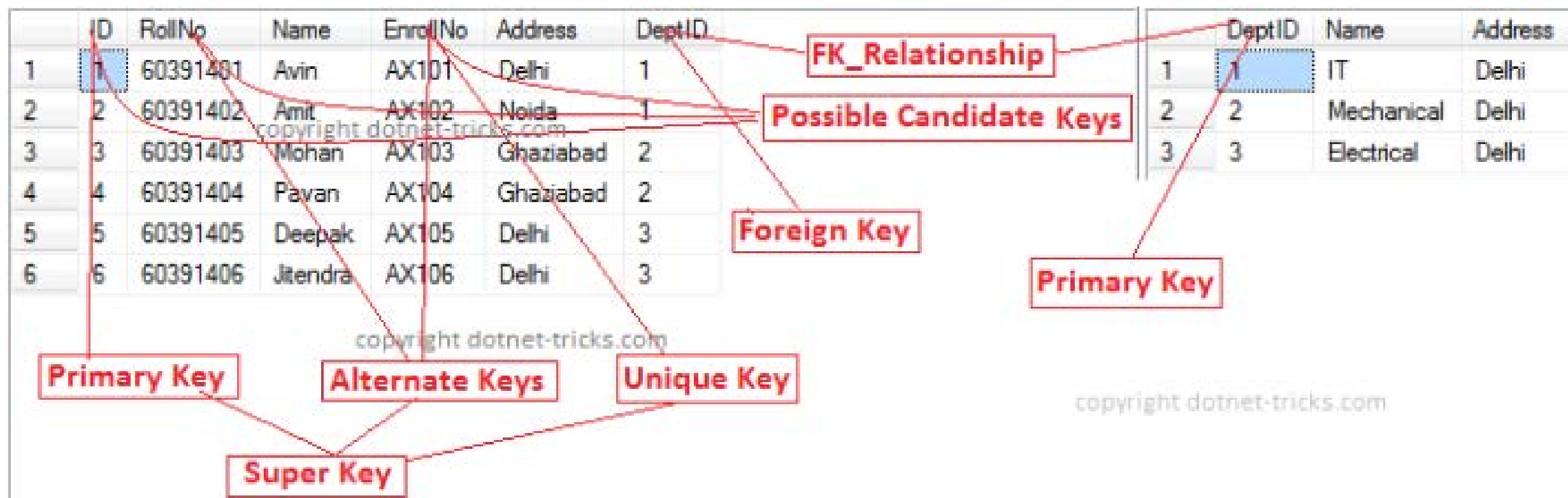
6. Unik nøgle

En unik nøgle er et sæt af et eller flere felter / kolonner af en tabel, der unikt identificerer en post i en database tabel. Det er som primærnøgle, men det kan kun acceptere én null værdi og det kan ikke have gentagne værdier.

7. Fremmed nøgle

En fremmed nøgle er et felt i database tabel, der er primær nøgle i en anden tabel. Det kan acceptere flere null og gentagne værdier.

Nøgle typer



Forbind data til ny sæt - teori

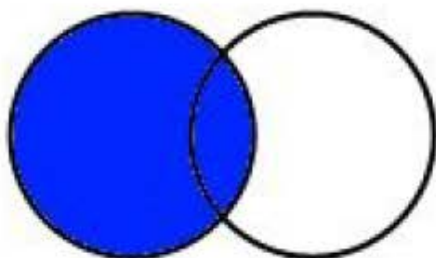
Fordi data sjældent ligger i kun én tabel

Hvad er Joins

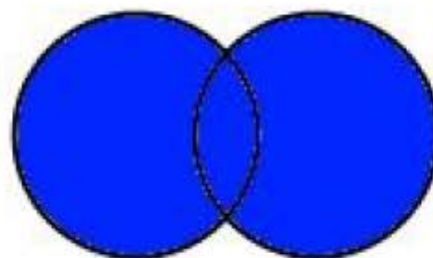
- Et SQL join kombinerer kolonner fra en eller flere tabeller i en relationel database.
- Det skaber et sæt, der kan gemmes som et table eller bruges som det er.
- Et JOIN er et middel til at kombinere kolonner fra en (selvstændig tilslutning) eller flere tabeller ved at bruge værdier, der er fælles for hver. ANSI-standard SQL angiver fem typer JOIN: INNER, LEFT OUTER, RIGHT OUTER, FULL OUTER og CROSS. Som et specielt tilfælde kan en tabel (basisbord, visning eller sammenføjet tabel) LØSES til sig selv i en selvforening.
- En programmør erklærer en JOIN-erklæring for at identificere rækker for tilslutning. Hvis det evaluerede prædikat er sandt, produceres den kombinerede række i det forventede format, en rækkesæt eller en midlertidig tabel.

Joins

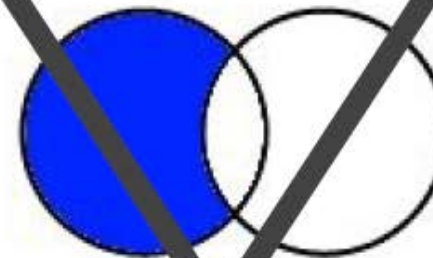
LEFT JOIN



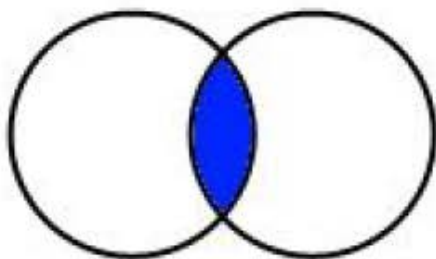
FULL OUTER JOIN



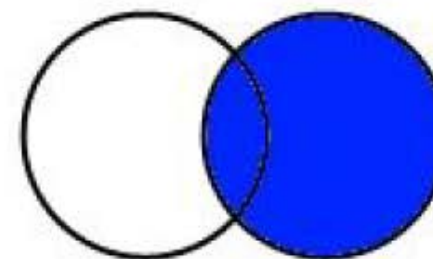
LEFT JOIN
(if NULL)



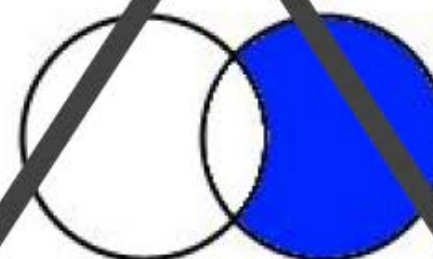
INNER JOIN



RIGHT JOIN



RIGHT JOIN
(if NULL)



Inner Join

```
SELECT * FROM tableA
```

```
JOIN tableB ON
```

```
tableA.tb_id =
```

```
tableB.id
```

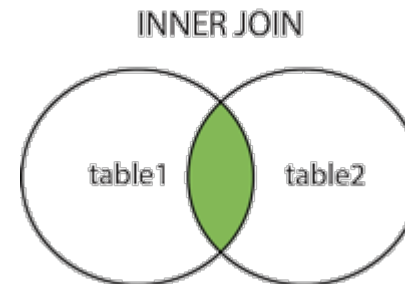
```
SELECT * FROM tableA
```

```
INNER JOIN tableB ON
```

```
tableA.tb_id =
```

```
tableB.id
```

id	name	id	name
--	----	--	----
1	Pirate	2	Pirate
3	Ninja	4	Ninja

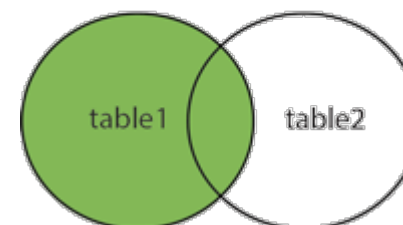


Left Join

```
SELECT * FROM tableA  
LEFT JOIN tableB ON  
tableA.tb_id =  
tableB.id
```

id	name	id	name
--	----	--	----
1	Pirate	2	Pirate
2	Monkey	null	null
3	Ninja	4	Ninja
4	Spaghetti	null	null

LEFT JOIN

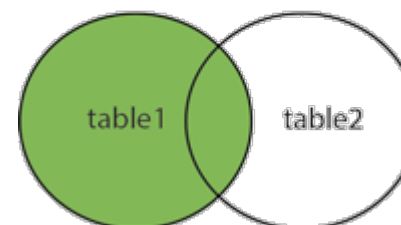


Right Join

```
SELECT * FROM tableA  
LEFT JOIN tableB ON  
tableA.tb_id =  
tableB.id
```

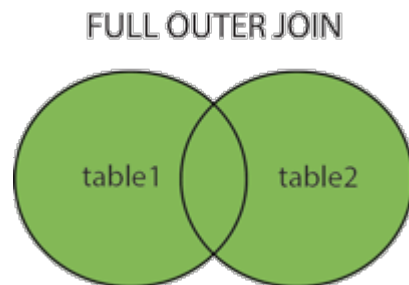
id	name	id	name
--	----	--	----
1	Pirate	2	Pirate
2	Monkey	null	null
3	Ninja	4	Ninja
4	Spaghetti	null	null

LEFT JOIN



Outer Join

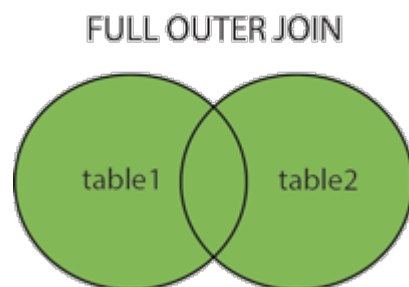
```
SELECT * FROM TableA
FULL OUTER JOIN TableB
ON TableA.name =
TableB.name
```



id	name	id	name
--	----	--	----
1	Pirate	2	Pirate
2	Monkey	null	null
3	Ninja	4	Ninja
4	Spaghetti	null	null
null	null	1	Rutabaga
null	null	3	Darth Vader

Outer Join

```
SELECT * FROM TableA
FULL OUTER JOIN TableB
ON TableA.name =
TableB.name
```

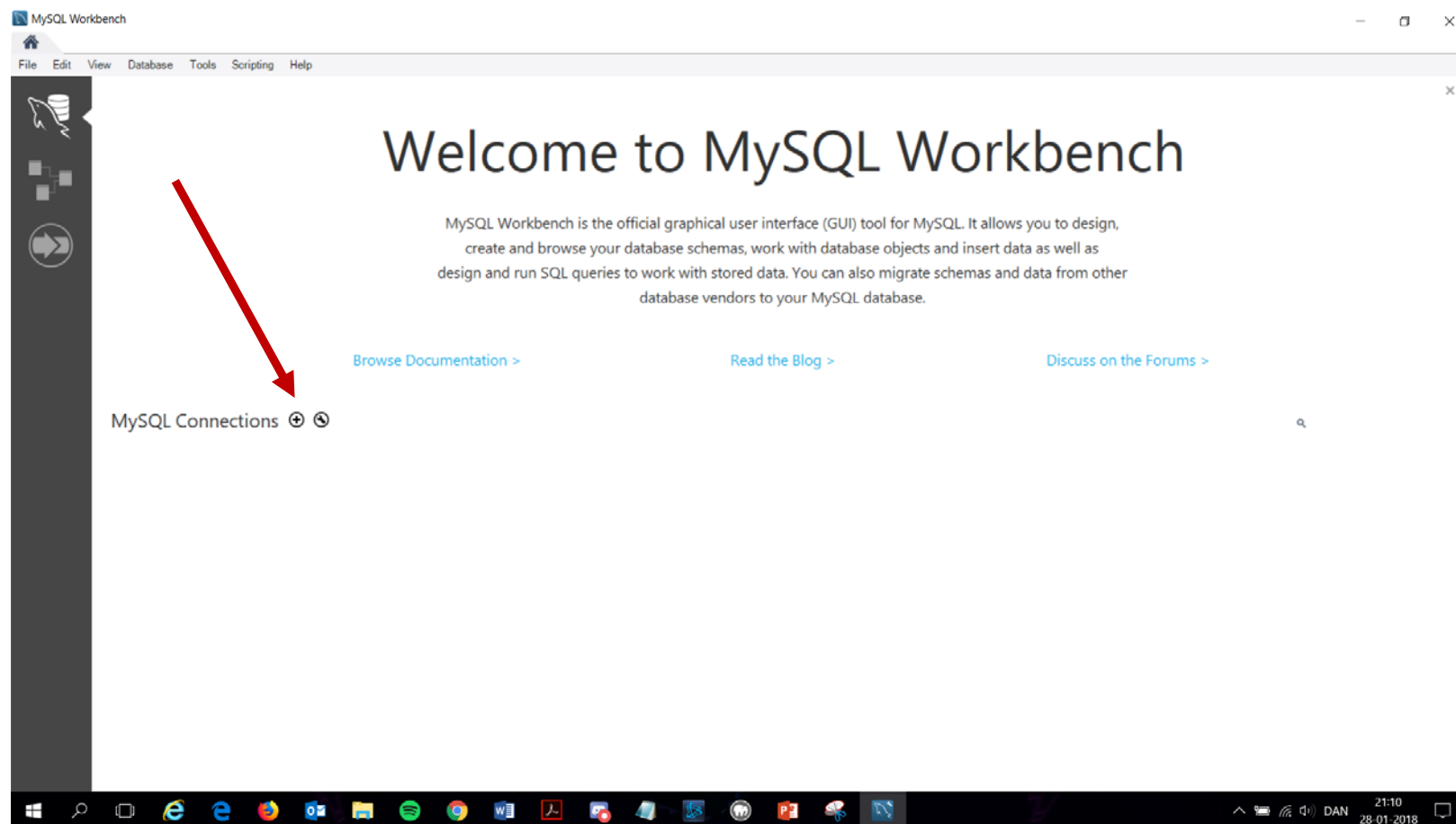


id	name	id	name
--	----	--	----
1	Pirate	2	Pirate
2	Monkey	null	null
3	Ninja	4	Ninja
4	Spaghetti	null	null
null	null	1	Rutabaga
null	null	3	Darth Vader

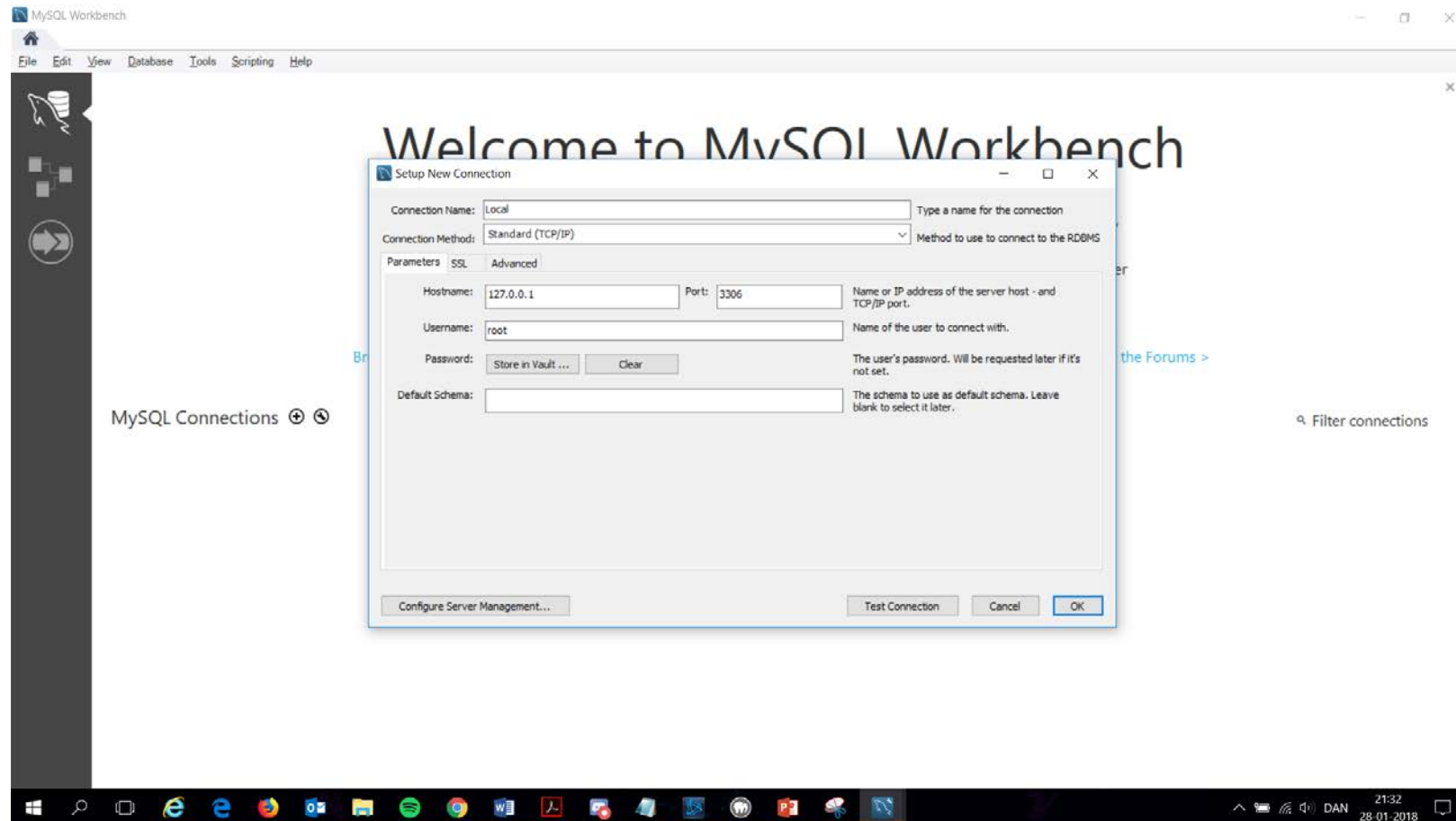
Forbind MySQL Workbench til MAMP

Forbind din workbench til din lokale database server

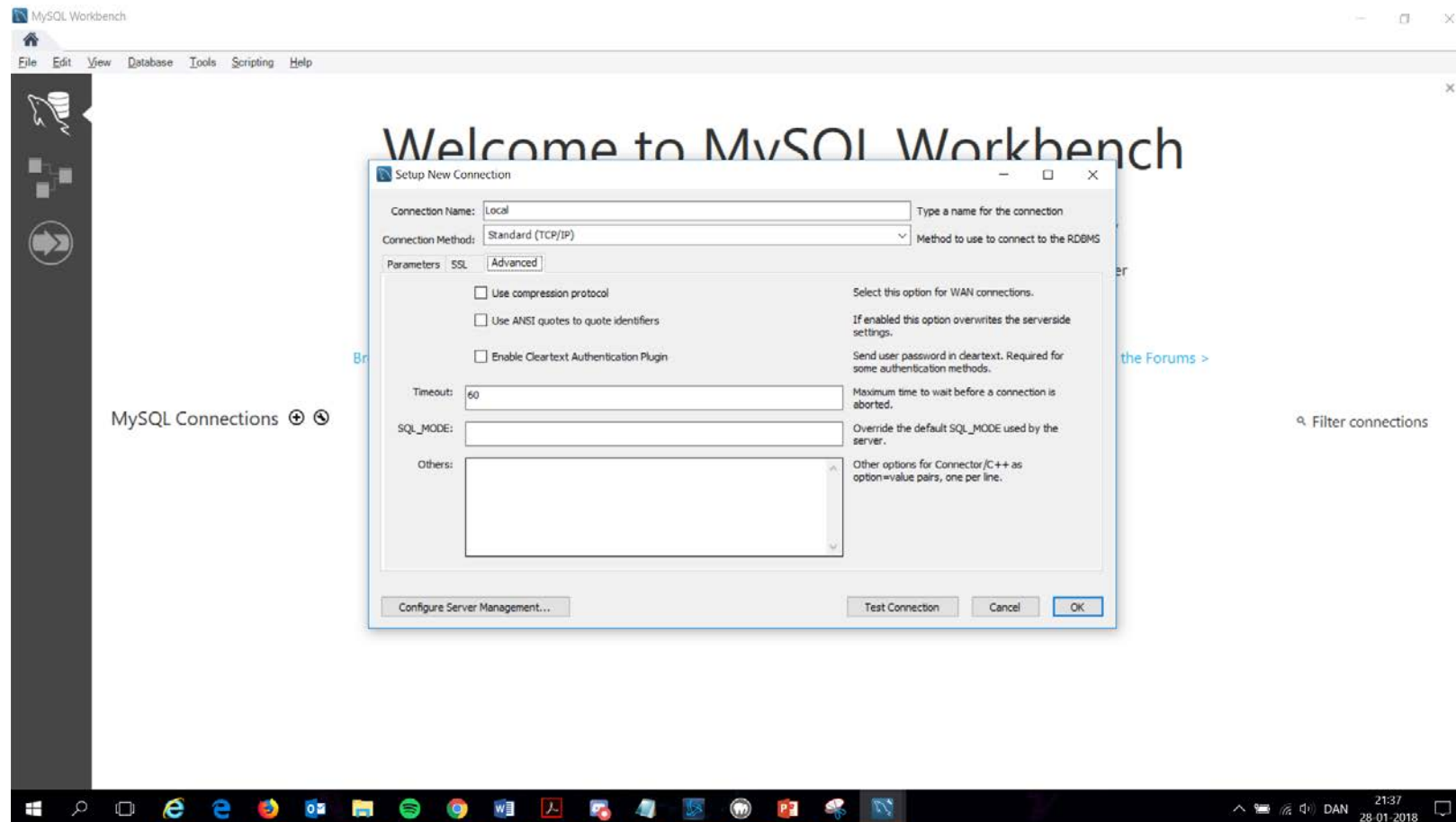
Lav en ny connection



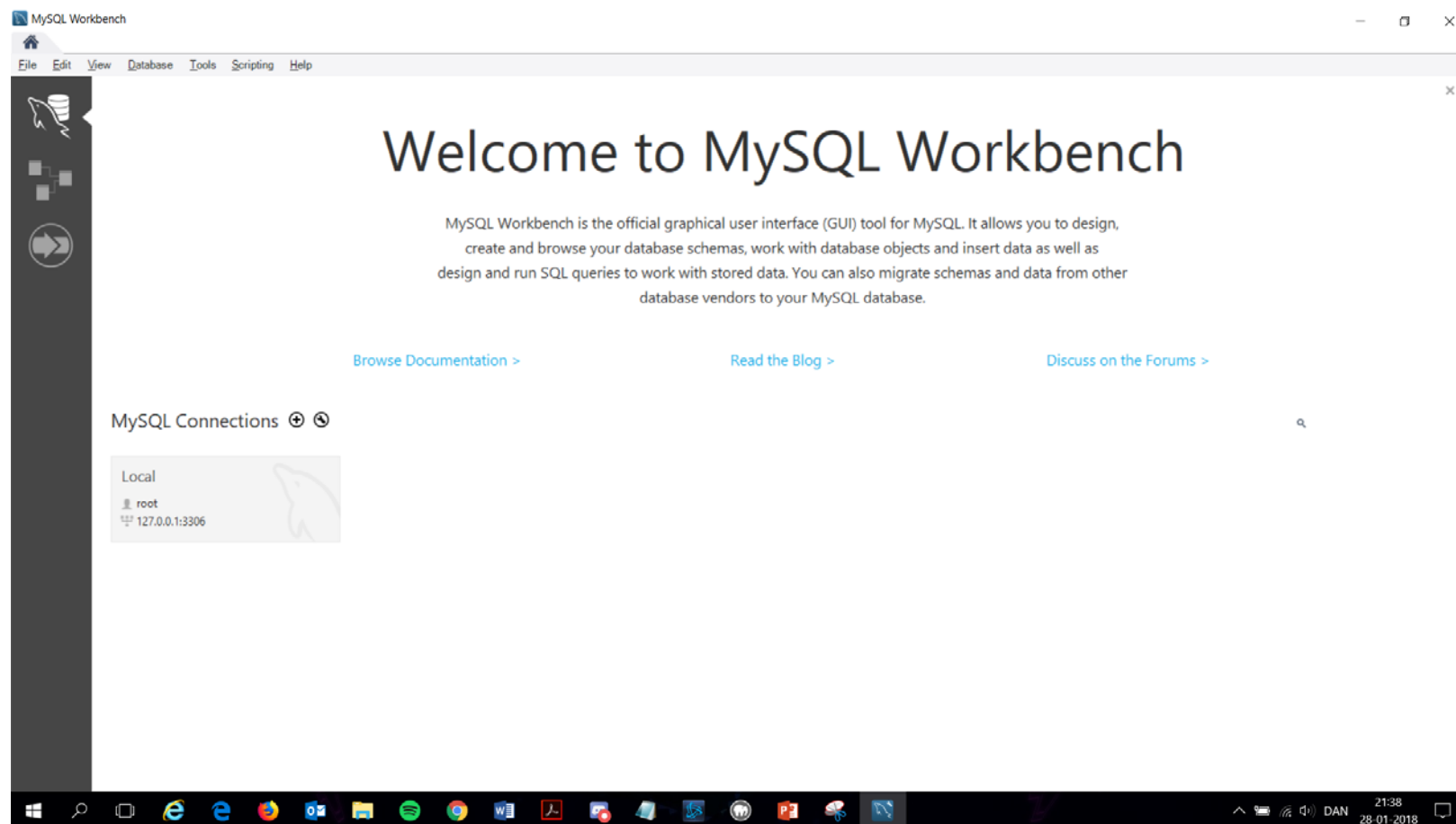
Sæt forbindelsen op, info herunder passer med default MAMP indstillinger



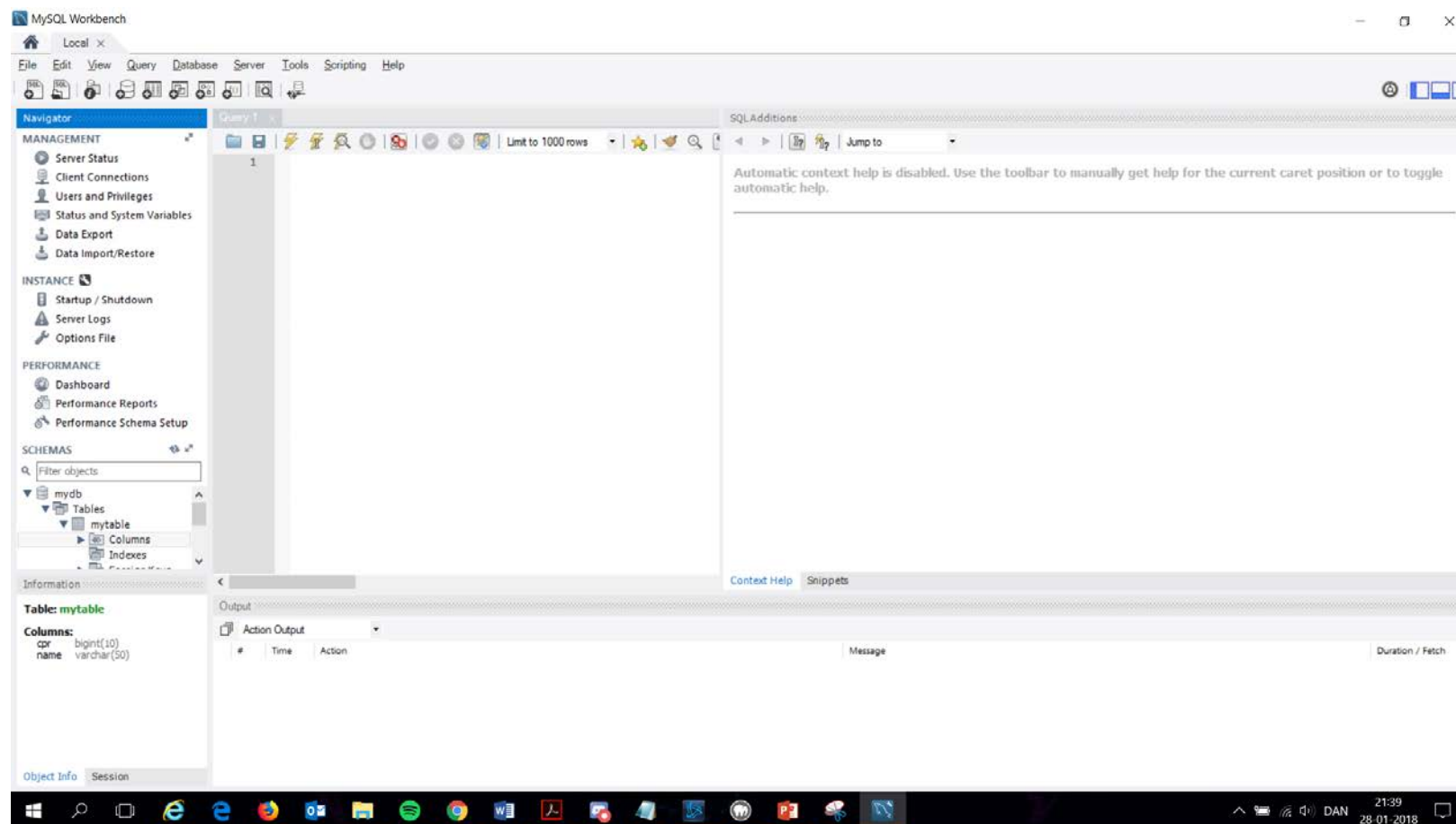
Hvis der er problemer så sikr at der ikke står noget under Advanced fanen



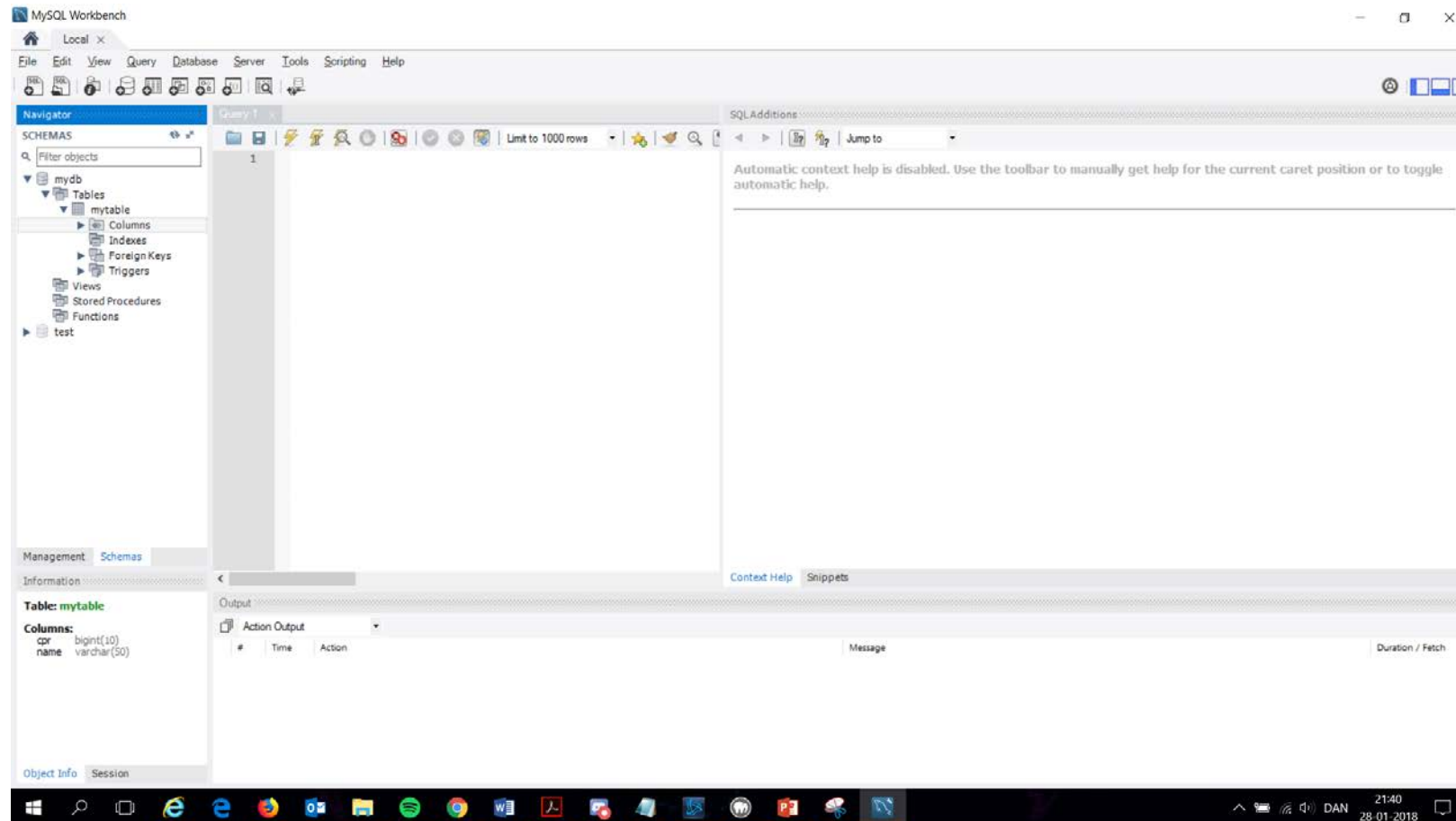
Nu er forbindelsen på start-skærmen



Vi kan nu se og arbejde direkte med vores databaser



Den dobbelt pil i hjørnet gør tabel listen større/mindre



Gennemgang af større eksempel

Hent flights.mwb og flights.sql fra denne lektions mappe på Fronter

Lektier til næste gang

Læsning

Læsning til næste gang

- Læs side 54 – 92 i SQL - The Ultimate Beginners Guide
- Lav SQL koden til at oprette den tabel du skitserede som diagram til denne lektion
- Læs om forskellene mellem en unik nøgle og en primær nøgle
<http://www.dotnettricks.com/learn/sqlserver/difference-between-primary-key-and-unique-key>
- Læs om forskellene mellem en primær og en fremmed nøgle
<http://www.dotnettricks.com/learn/sqlserver/difference-between-primary-key-and-foreign-key>

Kilder

Materiale benyttet i denne undervisningsgang

Kilder

Forskellige database systemer

- https://en.wikipedia.org/wiki/List_of_relational_database_management_systems
- <https://db-engines.com/en/ranking>
- <https://www.serverwatch.com/server-trends/slideshows/top-10-enterprise-database-systems-to-consider-2015.html>
- <https://www.cybrary.it/0p3n/database-differences-microsoft-sql-server-vs-oracle-database/>
- <https://db-engines.com/en/system/Microsoft+SQL+Server%3BMySQL>

Kilder

Den relationelle model

- Wilton, Paul & Colby, John. Beginning SQL. Wiley Publishing. 2005.
- https://en.wikipedia.org/wiki/Boyce–Codd_normal_form
- https://en.wikipedia.org/wiki/Oracle_Database
- https://en.wikipedia.org/wiki/Larry_Ellison
- https://www.tutorialspoint.com/dbms/relational_data_model.htm

Kilder

Normalisering uddybet

- <https://www.studytonight.com/dbms/database-normalization>
- <https://www.guru99.com/database-normalization.html>

Mere end CRUD

- ANSI/ISO/IEC International Standard (IS). Database Language SQL—Part 2: Foundation (SQL/Foundation). 1999
- https://en.wikipedia.org/wiki/Boolean_algebra
- <https://www.w3schools.in/mysql/ddl-dml-dcl/>
- https://www.w3schools.com/sql/sql_create_db.asp
- <http://www.tutorialspoint.com/mysql/mysql-whereclause.htm>
- <http://www.mysqltutorial.org/mysql-limit.aspx>
- <http://www.tutorialspoint.com//sql/sql-order-by.htm>

Kilder

Nøgler låser data op

- <http://www.dotnettricks.com/learn/sqlserver/different-types-of-sql-keys>
- <http://findnerd.com/list/view/Different-Types-of-SQL-Keys/21057/>

Forbind data til ny sæt

- https://www.w3schools.com/sql/sql_join.asp
- [https://en.wikipedia.org/wiki/Join_\(SQL\)](https://en.wikipedia.org/wiki/Join_(SQL))
- <http://blog.codinghorror.com/a-visual-explanation-of-sql-joins/>