Oblig3, oppgave 2e:  
Tabellen **Prøve** inneholder data om vannprøver som elever ved en ungdomsskole  
gjennomfører i vann/innsjøer i nærområdet:

− **Prøve**(prøvenr, enr, fornavn, dato, vid, vnavn, ph, temp)

Eksempelraden under viser at vannprøve 176 ble tatt av elev 24 (Kari), 24. august 2011 i vann nr. 5 (Pyttvann), og for denne prøven ble pH (surhetsgrad) målt til 5.6 og temperaturen til 13.2 grader Celcius:

(176, 24, 'Kari', '24.08.2011', 5, 'Pyttvann', 5.6, 13.2)

Gjør de forutsetningene du mener er nødvendige og rimelige. Skriv ned de funksjonelle avhengighetene, bestem kandidatnøkkel og normalform, og utfør deretter normalisering til 3NF.

**JSON**-objekt i **MongoDB**:

**>use vannprove;   
>db.dropDatabase();  
>db.createCollection("vann");  
>db.vann.deleteMany({});  
>db.vann.insert([** {"prøvenr": 176, "enr": 24, "fornavn": "Kari", "dato": "24.08.2011",   
 "vid": 5, "vnavn": "Pyttvann", "ph": 5.6, "temp": 13.2},  
 {"prøvenr": 181, "enr": 16, "fornavn": "Jan", "dato": "24.08.2011",   
 "vid": 2, "vnavn": "Steintjønn", "ph": 6.0, "temp": 16.5},  
 {"prøvenr": 153, "enr": 31, "fornavn": "Stein", "dato": "24.08.2011",   
 "vid": 3, "vnavn": "Sortsjøen", "ph": 5.8, "temp": 14.3},   
 {"prøvenr": 45, "enr": 24, "fornavn": "Kari", "dato": "23.08.2011",   
 "vid": 1, "vnavn": "Rokosjøen", "ph": 5.9, "temp": 16.0},  
 {"prøvenr": 18, "enr": 12, "fornavn": "Anne", "dato": "23.08.2011",   
 "vid": 5, "vnavn": "Pyttvann", "ph": 5.7, "temp": 13.5},   
 {"prøvenr": 83, "enr": 31, "fornavn": "Stein", "dato": "23.08.2011",   
 "vid": 3, "vnavn": "Sortsjøen", "ph": 5.7, "temp": 15.5},  
 {"prøvenr": 43, "enr": 37, "fornavn": "Inger", "dato": "23.08.2011",  
 "vid": 3, "vnavn": "Sortsjøen", "ph": 5.7, "temp": 15.5},  
 {"prøvenr": 77, "enr": 47, "fornavn": "Gunnar", "dato": "25.08.2011",  
 "vid": 5, "vnavn": "Pyttvann", "ph": 5.8, "temp": 16.3},  
 {"prøvenr": 8, "enr": 24, "fornavn": "Kari", "dato": "24.08.2011",  
 "vid": 3, "vnavn": "Sortsjøen", "ph": 5.9, "temp": 15.8},  
 {"prøvenr": 123, "enr": 47, "fornavn": "Gunnar", "dato": "25.08.2011",   
 "vid": 6, "vnavn": "Gaukilsjøen", "ph": 5.5, "temp": 13.8},  
 {"enr": 37, "fornavn": "Unni" },  
 {"enr": 27, "fornavn": "Roar" },  
 {"enr": 38, "fornavn": "Milli" },  
 {"vid": 7, "vnavn": "Sætersjøen" } **]);**

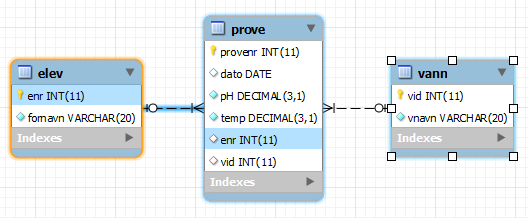
I kommandovindu: **mongo use oblig3 >db.createCollection("vann"); db.vann.deleteMany({}); db.vann.drop();**  
Som **relasjon**, altså tabell med restriksjon om entydighet, primærnøkkel:  
**CREATE TABLE vann (  
 provenr INT,  
 enr INT, #evt NOT NULL, UNIQUE   
 fornavn VARCHAR(20) NOT NULL,**  
 **dato DATE,  
 vid INT, #evt NOT NULL, UNIQUE  
 vnavn VARCHAR(20) NOT NULL,  
 pH DECIMAL(3,1) NOT NULL,  
 temp DECIMAL(3,1) NOT NULL  
);**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| prøvenr | enr | fornavn | dato | vid | vnavn | ph | temp |
| 176 | 24 | Kari | 24.08.2011 | 5 | Pyttvann | 5.6 | 13.2 |

Tabellen oppfyller **1NF**.   
**2NF** kommer ikke inn her siden vi ikke har en sammensatt primærnøkkel.  
Funksjonelle avhengigheter:  
 -**provenr** → alle kolonner  
 -**enr** → fornavn  
 -**vid** → vnavn  
De to siste avhengighetene bryter **3NF** siden primærnøkkel ikke er med (de er da transitive avhengigheter siden **enr** og **vid** allerede avhenger av **provenr**).  
Vokabular: supernøkkel, kandidatnøkkel, determinant.

Bryter opp tabellen slik at **3NF** blir oppfylt:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **elev** | |  | **vann** | |  | **prove** | | | | | | |
| enr | fornavn | vid | vnavn | prøvenr | enr | dato | vid | ph | temp | prøvenr |
| 24 | Kari | 5 | Pyttvann | 176 | 24 | 24.08.2011 | 5 | 5.6 | 13.2 | 176 |

 **DROP DATABASE IF EXISTS Vannprøve;  
CREATE DATABASE IF NOT EXISTS Vannprøve;  
USE Vannprøve;  
CREATE TABLE elev (  
 enr INT PRIMARY KEY,  
 fornavn VARCHAR(20) NOT NULL  
) ENGINE=InnoDB;  
CREATE TABLE vann (  
 vid INT PRIMARY KEY,  
 vnavn VARCHAR(20) NOT NULL  
) ENGINE=InnoDB;  
CREATE TABLE prove (  
 provenr INT,  
 dato DATE,  
 pH DECIMAL(3,1) NOT NULL,  
 temp DECIMAL(3,1) NOT NULL,  
 enr INT,  
 vid INT,  
 CONSTRAINT PRIMARY KEY(provenr),  
 CONSTRAINT FOREIGN KEY(enr) REFERENCES elev(enr),  
 CONSTRAINT FOREIGN KEY(vid) REFERENCES vann(vid)  
) ENGINE=InnoDB;**

**DELETE FROM prove;  
DELETE FROM elev;  
DELETE FROM vann;  
INSERT INTO elev VALUES (24, 'Kari'), (16, 'Jan') , (31, 'Stein') , (12, 'Anne') , (37, 'Inger') ,   
 (47, 'Gunnar') , (21, 'Unni') , (38, 'Milli') , (27, 'Roar');  
INSERT INTO vann VALUES (5, 'Pyttvann'), (2, 'Steintjønn') , (3, 'Sortsjøen') , (1, 'Rokosjøen') , (6, 'Gaukilsjøen') ,   
 (7, 'Sætersjøen');  
INSERT INTO prove VALUES  
 (176, '2011-08-24', 5.6, 13.2, 24, 5),  
 (181, '2011-08-24', 6.0, 16.5, 16, 2),   
 (153, '2011-08-24', 5.8, 14.3, 31, 3),  
 ( 45, '2011-08-23', 5.9, 16.0, 24, 1),  
 ( 18, '2011-08-23', 5.7, 13.5, 12, 5),   
 ( 83, '2011-08-23', 5.7, 15.5, 31, 3),  
 ( 43, '2011-08-23', 5.7, 15.5, 37, 3),  
 ( 77, '2011-08-25', 5.5, 16.3, 47, 5),  
 ( 8, '2011-08-24', 5.9, 15.8, 24, 3),  
 (123, '2011-08-25', 5.5, 13.8, 47, 6);**

Skriv alle verdier i databasen **>db.vann.find().pretty();   
SELECT P.\*, E.\*, V.\* FROM prove AS P, elev AS E, vann AS V  
 WHERE P.enr=E.enr AND P.vid = V.vid;** (får dubletter, mangler elever uten prøver, vann uten prøver) **SELECT P.provenr, P.dato, V.vnavn, P.ph, P.temp, E.fornavn FROM prove AS P, elev AS E, vann AS V  
 WHERE P.enr=E.enr AND P.vid = V.vid;** (ingen dubletter, mangler elever uten prøver, vann uten prøver) **SELECT P.provenr, P.dato, V.vnavn, P.ph, P.temp, E.fornavn FROM prove AS P JOIN elev AS E JOIN vann AS V  
 ON P.enr=E.enr AND P.vid = V.vid;  
SELECT P.provenr, P.dato, P.ph, P.temp, E.fornavn   
 FROM (elev AS E LEFT OUTER JOIN prove AS P ON P.enr=E.enr);** (får med elever uten prøver)  
**SELECT P.provenr, P.dato, P.ph, P.temp, V.vnavn   
 FROM (prove AS P RIGHT OUTER JOIN vann AS V ON P.vid=V.vid);** (får med vann uten prøver)  
**SELECT P.provenr, P.dato, P.ph, P.temp, E.fornavn, V.vnavn   
 FROM (elev AS E LEFT OUTER JOIN prove AS P ON P.enr=E.enr)  
 RIGHT OUTER JOIN vann AS V ON P.vid=V.vid;** (??)

Skriv alle verdier i databasen sortert stigende etter fornavn **>db.vann.find().sort({"fornavn":1}).pretty();   
SELECT P.provenr, P.dato, V.vnavn, P.ph, P.temp, E.fornavn FROM (prove AS P JOIN elev AS E ON P.enr=E.enr)   
 JOIN vann AS V ON P.vid = V.vid ORDER BY E.fornavn;**

Finn antall registreringer **>db.vann.find().count();  
SELECT COUNT(\*) FROM prove;**

Skriv alle målinger foretatt av Kari **>db.vann.find({"fornavn":"Kari"}).pretty();   
SELECT P.provenr, P.dato, V.vnavn, P.ph, P.temp, E.fornavn FROM (prove AS P JOIN elev AS E ON P.enr=E.enr)   
 JOIN vann AS V ON P.vid = V.vid WHERE E.fornavn='Kari';**

Skriv alle målinger foretatt den 23. august **>db.vann.find({"dato":"23.08.2011"}).pretty();   
SELECT P.provenr, P.dato, V.vnavn, P.ph, P.temp, E.fornavn FROM (prove AS P JOIN elev AS E ON P.enr=E.enr)   
 JOIN vann AS V ON P.vid = V.vid WHERE P.Dato='2011-08-23';**Finn alle målinger der pH>5.7  
**>db.vann.find({"ph":{$gt:5.7}}).pretty();  
SELECT P.provenr, P.dato, V.vnavn, P.ph, P.temp, E.fornavn FROM (prove AS P JOIN elev AS E ON P.enr=E.enr)   
 JOIN vann AS V ON P.vid = V.vid WHERE P.ph>5.7;**

Skriv ut alle elevnavnene **>db.vann.aggregate({$project:{"\_id":0,"fornavn":1}});**  
NB: kan ikke koble "aggregate" med "distinct" i MongoDB **>db.vann.distinct("fornavn");  
SELECT fornavn FROM elev;  
SELECT DISTINCT E.fornavn FROM prove AS P JOIN elev AS E ON P.enr=E.enr;** (uten prøveløse elever)Finn antall målinger hver elev har foretatt **>db.vann.aggregate({$group:{"\_id":"$dato", antall:{$sum:1}}});** (for enkelt, tar med elevere uten målinger)  
**>** **db.vann.aggregate([  
 {$match:{ph:{$gte:0}}},** (bare elever med målinger telles med) **{$group:{"\_id":"$fornavn", antall:{$sum:1}}}  
]);  
SELECT E.fornavn, COUNT(P.enr) AS antall FROM prove AS P   
 RIGHT OUTER JOIN elev AS E ON P.enr=E.enr   
 GROUP BY P.enr;**

Finn antall målinger foretatt i hvert vann **>db.vann.aggregate({$group:{"\_id":"$vnavn", antall:{$sum:1}}});** (tar med sjøer uten målinger) **>db.vann.aggregate([  
 {$match:{ph:{$gte:0}}},   
 {$group:{"\_id":"$vnavn", antall:{$sum:1}}}  
]);** (bare vann med målinger telles med) **SELECT V.vnavn, COUNT(P.vid) AS antall FROM prove AS P   
 RIGHT OUTER JOIN vann AS V ON P.vid=V.vid   
 GROUP BY P.vid;**

Finn elever som ikke har tatt prøver  
**>db.vann.find({vid:{$exists:false}},{\_id:0, fornavn:1});  
SELECT E.fornavn, E.enr FROM elev AS E where E.enr NOT IN (SELECT enr FROM prove);**

Finn alle elever og alle prøver av de elevene som har tatt prøver  
**>db.vann.aggregate({$project:{"\_id":0,"fornavn":1, "vnavn":1}});  
>db.vann.find({}, {"\_id":0,"fornavn":1, "vnavn":1});  
SELECT E.fornavn, 'har målt pH i', V.vnavn FROM (elev E JOIN prove P   
 ON E.enr = P.enr) RIGHT JOIN Vann V ON V.vid = P.vid;**

Du vil legge inn nye data:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| prøvenr | enr | fornavn | dato | vid | vnavn | ph | temp |
| 91 | 88 | Khalid | 23.08.2011 | 8 | Harasjøen | 5.9 | 16.7 |

og har begynt å skrive en prosedyre for dette.   
Den ser foreløpig ut som  
DELIMITER //  
CREATE PROCEDURE SettInn(  
 IN provenr INT, IN enr INT, IN fornavn VARCHAR(45), IN dato DATE,   
 IN vid INT, IN vnavn VARCHAR(45) , IN ph DECIMAL(4,2) , IN temp DECIMAL(5,2)  
)  
BEGIN  
 **START TRANSACTION;**  
 INSERT INTO elev VALUES (enr, fornavn);  
 INSERT INTO vann VALUES (vid, vnavn);  
 INSERT INTO prove VALUES (provenr, dato, ph, temp, enr, vid);  
 **COMMIT;**  
END //  
DELIMITER ;  
Hvordan skal en slik prosedyre eksekveres?  
CALL SettInn(91, 88, 'Khalid', '2011.08.23', 8, 'Harasjøen', 5.9, 16.7);

**Web-side i vannprove:  
$setning = "SELECT fornavn FROM elev ORDER BY fornavn";**



**$setning = "SELECT P.provenr, P.dato, V.vnavn, P.ph, P.temp, E.fornavn  
 FROM (prove AS P JOIN elev AS E ON P.enr=E.enr)  
 JOIN vann AS V ON P.vid = V.vid WHERE E.fornavn='$fornavn'";**

Et bilde som inneholder bord

Automatisk generert beskrivelse