

# Data analyse – Business Keys

Vejen til en RDBMS database

## Oplæg

Dette er den første løsningsmodel hvor vi sigter mod målet at opretholde keys som business keys. I denne løsning er der ingen '\_id' kolonner og nøgler realiseret ved automatisk inkrementering. Vi anvender straight business keys hvilket giver en del redundans i relaterede tabel, men det er prisen.

I et opfølgende oplæg løser vi opgaven med auto id kolonner.

- Det indledende om dataanalyse er selvfølgelig det samme uanset hvilken model man vælger at realisere.
- Men der er ret stor forskel når det gælder implementeringen, auto nummereringen tilføjer kompleksitet fordi vi når vi fylder tabeller med data, har behov for at kunne slå allerede oprettede rækker og deres id'er op, når vi fylder data med referentiel integritet over auto id kolonner.
- og der er fordele og ulemper ved begge tilgange.

Tendensen bevæger sig mod auto id løsninger, fordi det feks. kan være en forudsætning for at anvende frameworks og kodegeneratorer. I en auto id løsning kan man gennem en constraint stadig opretholde en forventning om unikke værdier. Det er tit også nemmere at migrere data i auto id databaser.

## Forhistorie

Vi har indhentet data, og vi har etableret følgende databaser

### Cars\_stage:

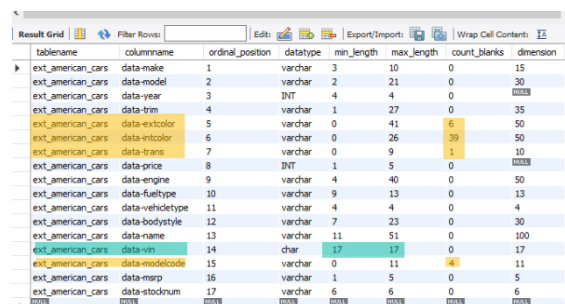
En database vi bruger til at transformere data. Indlæsning og derefter typecasting samt hvis der havde været det behov, feks. afledte kolonner. Indlæsningstabeller hedder '**ext**\_**<tabelnavn>**', ext for extract. Typed data findes i tabellen '**dat**\_**<tabelnavn>**', dat for data.

Database objekter der anvendes til transformering er navngivet som '**tfm**\_**<type>**\_**<tabelnavn>**' (et view og en stores procedure) pr. tabel. Nogle integrationer kunne godt indeholde flere tabeller, feks. kan man i et MS Excel dokument gemme flere faneblade. Hvert faneblad vil aflede behovet for en tabel. En anden form kunne være en zip fil.

### Cars\_template:

En database der genererer databaseobjekter i feks. cars\_stage databasen. Man kunne have andre databaser med helt andre typer oplysninger, denne database vil md få tilpasninger kunne anvendes til etablering af integrationer på tværs af databaser, men så skulle databasen nok også have et andet navn...

Tabellen template\_meta indeholder meta data om data, som vi fandt dem da vi skrabede bilforhandlerens webside. Der er mange bilmærker og modeller, men næppe alle varianter der findes på markedet. Der er de varianter han havde udstille på sin hjemmeside, da vi skrabede siden.



tablename	columnname	ordinal_position	datatype	min_length	max_length	count_blanks	dimension
ext_american_cars	data-make	1	varchar	3	10	0	15
ext_american_cars	data-model	2	varchar	2	21	0	30
ext_american_cars	data-year	3	INT	4	4	0	1000
ext_american_cars	data-trim	4	varchar	1	27	0	35
ext_american_cars	data-extcolor	5	varchar	0	41	6	50
ext_american_cars	data-entcolor	6	varchar	0	26	39	50
ext_american_cars	data-trans	7	varchar	0	9	1	10
ext_american_cars	data-price	8	INT	1	5	0	1000
ext_american_cars	data-engine	9	varchar	4	40	0	50
ext_american_cars	data-fueltype	10	varchar	9	13	0	13
ext_american_cars	data-vehedtype	11	varchar	4	4	0	4
ext_american_cars	data-bodystyle	12	varchar	7	23	0	30
ext_american_cars	data-name	13	varchar	11	51	0	100
ext_american_cars	data-vin	14	char	17	17	0	17
ext_american_cars	data-modelcode	15	varchar	0	11	4	11
ext_american_cars	data-msrp	16	varchar	1	5	0	5
ext_american_cars	data-stocknum	17	varchar	6	6	0	6

# Data analyse – Business Keys

## Vejen til en RDBMS database

Ikke desto mindre vælger vi at sætte vores lid til det vi har fundet, måske vælger vi at gøre vores system lidt mere robust, feks. hvis vi forventer et tekstfelt ret sandsynligt kan rumme flere karakterer ind hvad vi lige ser repræsenteret (ved at selectere distinct på kolonnen vi interesserer os for). Af samme grund er vores extract tabel dimensioneret så den kan rumme meget længere tekststreng. Strategien er at indlæse data fra fysisk fil kompromisløst. Ingen begrænsninger, ingen datatyper der ikke kan tåle bogstaver, ingen nøgler. Smidigheden i vores værktøj som cars\_stage databasen skal betragtes som, giver os muligheder for agilt hurtigt at kunne reetablere integrationen ved at danne nye tfm objekter, der virker efter hensigten efter vores nye viden om data.

## Tilgange til data analyse

### Finde nøgler

Template\_meta har en kolonne der hedder 'count\_blanks'. Igen med udgangspunkt i de data vi kunne se, tælles der antal forekomster hvor der ikke er oplyst data. Umiddelbart kan vi med det samme afskrive disse kolonner som nøglekandidat.

Til gengæld er kolonnen data-vin interessant, alle forekomster er 17 karakterer lange, og data kunne godt ligne stelnumre. Så med denne kan vi identificere de enkelte forekomster af biler der stod til salg.

data-name	data-vin	data-modelcode	data-msrp	data-stocknum
2004 Ford F-150 Lightning	2T7W073H4CA16212	F07	30999	US4045
2008 Ford Super Duty F-350 DRW XL Open Ser...	1FD4W3R61RED13504	W57	18999	US4891
2013 Ford Escape Titanium 4WD	1PMCU5J91DUA69678	U9J	16999	US5149
2014 Toyota Corolla S	2T1BURHE1EC083842	1862	19999	US5188
2014 Nissan Rogue S AWD	5N1AT2MVECB34551	22214	18999	US5240
2014 Ford Fusion SE	1FA6P0R46R172737	RM	0	US5290
2015 Jeep Wrangler Unlimited Sport 4WD	1C4B3WGP5L444411	JCM74	28999	US4936
2015 Ford F-150 Platinum SuperCrew	1FTEW1EG7FA07940	W1E	34999	US5191
2015 Jeep Wrangler Unlimited Sport 4WD	1C4B3WGP5L609533	JCM74	28999	US5212
2015 Hyundai Sonata 2.4L SE	SHPE24AFBPH008319	28402F45	15999	US5278
2015 Jeep Wrangler Rubicon Hard Rock 4WD	1C4H1JWC0GL157886	JCS72	38999	US4754
2016 Honda Civic Sedan EX-T	2H5FC3F3WGH655512	FC3F3C3W	24999	US4915

Vi kan afprøve vores tese med feks en sql som denne:

```
select count(distinct `data-vin`), count(`data-vin`)
from cars_stage.ext_american_cars;
```

Forespørgelsen returnerer antallet af disticte forekomster og antallet af forekomster i det hele taget. Hvis disse tal er ens, er data i kolonnen UNIKKE og det vil opfylde kriteriet for at være en nøgle.

### Brug din fornuft og kendskab til problemområdet

Kolonnoverskrifterne bør selvfølgelig på en kort og præcis og forståelig måde beskrive data indholdet. Data i sig selv kan du genkende, hvis du har domæne viden.

### Finde domæner og subdomæner

Når vi når hertil ved vi godt hvad der står i de enkelte kolonner. Vi ved sikkert også alle at man til næsten ethvert bilmærke kan få flere modeller, så det kan være vores første to tabeller.

Med de resterende kan overvejelserne være hvordan forholdet mellem engine og make/model mon forholder sig. Men det er nok sådan, at man ikke til enhver model kan få enhver motor. En V8 motor i en Ford Mustang kan ikke uden store modifikationer sidde i en Ford Ka. Så lad nu engine stå i forhold til model.

Nogle kolonner indeholder data som ikke på nogen måde relaterer til make-model. Transmission feks. er tydeligtvis forhandlerens egen værdilister. De data vi ikke kan relatere lægger vi i datalister, og de vil relatere sig direkte til det enkelte køretøj.

# Data analyse – Business Keys

Vejen til en RDBMS database

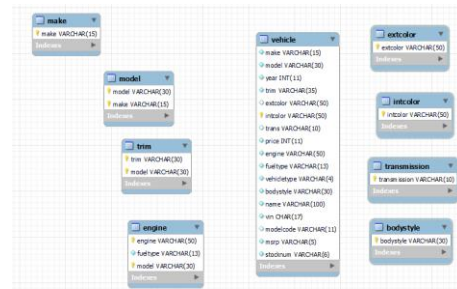
## Etablering af databasen

*cars\_rdbms\_business\_keys.sql*

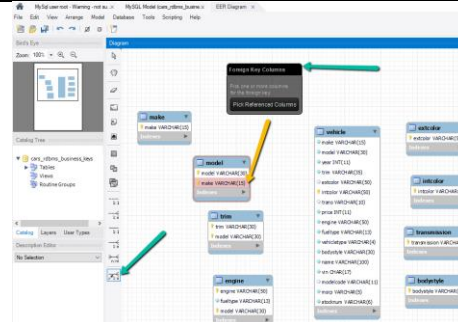
Man kan lægge ud med lavet tabellerne med business entiteter som er nødvendige for at etablere relationen.  
Model står i forhold til make, ved at have make kolonnen i model, kan vi bruge make som fremmed nøgle i model.

```
USE `cars_rdbms_business_keys`;  
/* DDL related table begins here */  
CREATE TABLE `make` (  
  `make` varchar(15) NOT NULL,  
  PRIMARY KEY (`make`)  
)ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
CREATE TABLE `model` (  
  `model` varchar(30) NOT NULL,  
  `make` varchar(15) NOT NULL,  
  PRIMARY KEY (`make`,`model`)  
)ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

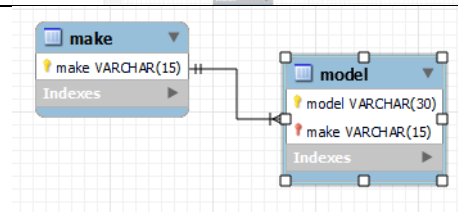
ER diagrammet med nøgler uden relationer



Lav en relation med ikonet nederst til venstre, og se dialog boksen der dukker op. Følg anvisningerne:



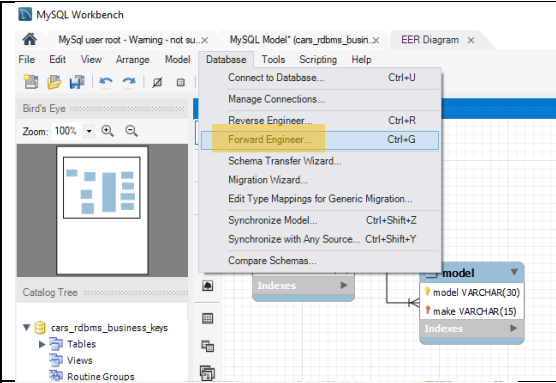

Og du har en en til mange relation mellem make og model



# Data analyse – Business Keys

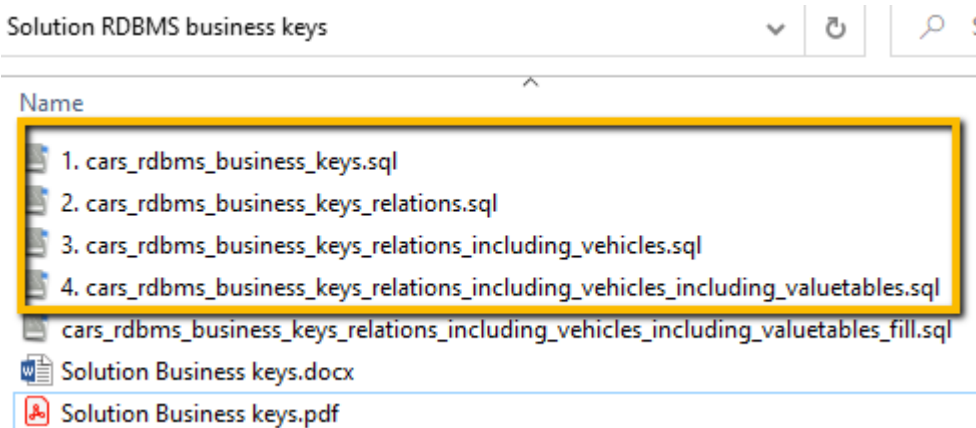
Vejen til en RDBMS database

## Next step

<p>Næste step kunne så være at få genereret DDL ud fra modellen med <b>Forward Engineer</b>.</p>	 A screenshot of the MySQL Workbench interface. The 'Database' menu is open, and the 'Forward Engineer' option is highlighted. The background shows a catalog tree with a database named 'cars_rdbms_business_keys' and an EER diagram.
<p>Følg dialog rækkefølgen i Forward Engineer to Database guiden, undervejs kan du se DDL kode som viser hvordan relationen er blevet etableret med DDL SQL</p> <p>NB i en af dialogerne vælger man om man vil droppe eksisterende objekter. Default var ikke at overskrive, og så bliver databaen selvfølgelig ikke opdateret...</p>	 A screenshot of the 'Forward Engineer to Database' dialog box. The 'Review the SQL Script to be Executed' tab is active, showing a SQL script that creates tables and relationships. The script includes comments and SQL syntax for creating 'cars_rdbms_business_keys', 'model', and 'transmission' tables with their respective constraints.

Følg hvordan opbygningen af databasen sker iterativt og vi gemmer hvert Forward Engineer script for hver iteration.

Solution RDBMS business keys

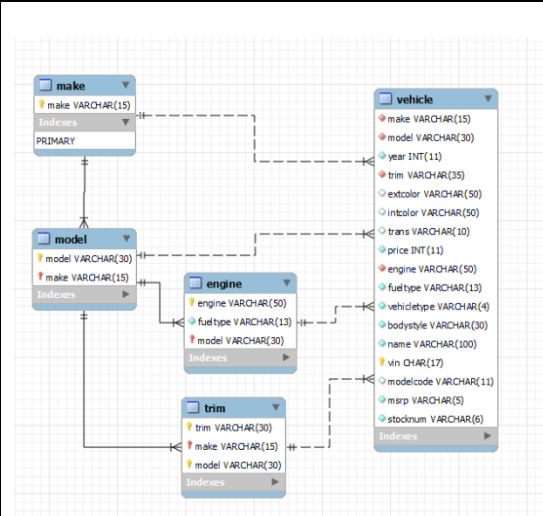
A screenshot of a file explorer window showing a list of files. The files are: 1. cars\_rdbms\_business\_keys.sql, 2. cars\_rdbms\_business\_keys\_relations.sql, 3. cars\_rdbms\_business\_keys\_relations\_including\_vehicles.sql, 4. cars\_rdbms\_business\_keys\_relations\_including\_vehicles\_including\_valuetables.sql, cars\_rdbms\_business\_keys\_relations\_including\_vehicles\_including\_valuetables\_fill.sql, Solution Business keys.docx, and Solution Business keys.pdf. The first four files are highlighted with a yellow box.

# Data analyse – Business Keys

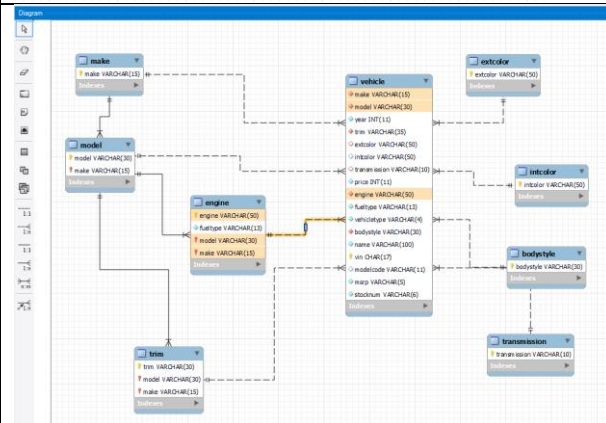
Vejen til en RDBMS database

## Iterationer

Måske foretrækker du code first tilgangen eller du foretrækker at arbejde videre i modellen (model first). Du kan bruge modellen og få genereret kode med Forward Engineer, eller du kan arbejde med koden og se resultatet af din indsats i ER diagrammet ved at bruge Reverse Engineer.



Her inkluderer vi værdilisterne og vores model er komplet



## Data fill

cars\_rdbms\_business\_keys\_relations\_including\_vehicles\_including\_valuetables\_fill.sql

Nu skal vi hælde data i vores database

Fordi vi bruger business keys er fill statements relativt ukompliceret

Med Schema Inspektoren kan du i fanen Tables se rækkeantal for hver tabel. Der skulle jo gerne være rækker i alle tabeller

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

Table	Rows
bodystyle	10
engine	10
extcolor	10
intcolor	10
make	10
model	10
transmission	10
trim	10
vehicle	10

# Data analyse – Business Keys

Vejen til en RDBMS database

**Ved som det endelige step, at hælde alle vores rækker fra cars\_stage.dat\_american\_cars til vores vehicle tabel, og gøre det uden fejl, er beviset for at vores relationer er intakte jf. modellen og scriptet.**

## Opmærksomhedspunkter

- Bemærk at kolumnen data-trans er oversat til transmission
- At DDL for vehicle tabeææen er identisk med DDL for cars\_stage.dat\_american\_cars
  - Især at for kolonnerne feks. extcolor, intcolor gælder at de er nullable i vehicles tabellen – empty values accepteres
- At vi slet og ret importerer data til vores vehicle tabel med en plain select på cars\_stage.dat\_american\_cars
- At fill er lige ud af landevejen, fordi vi arbejder med business keys.