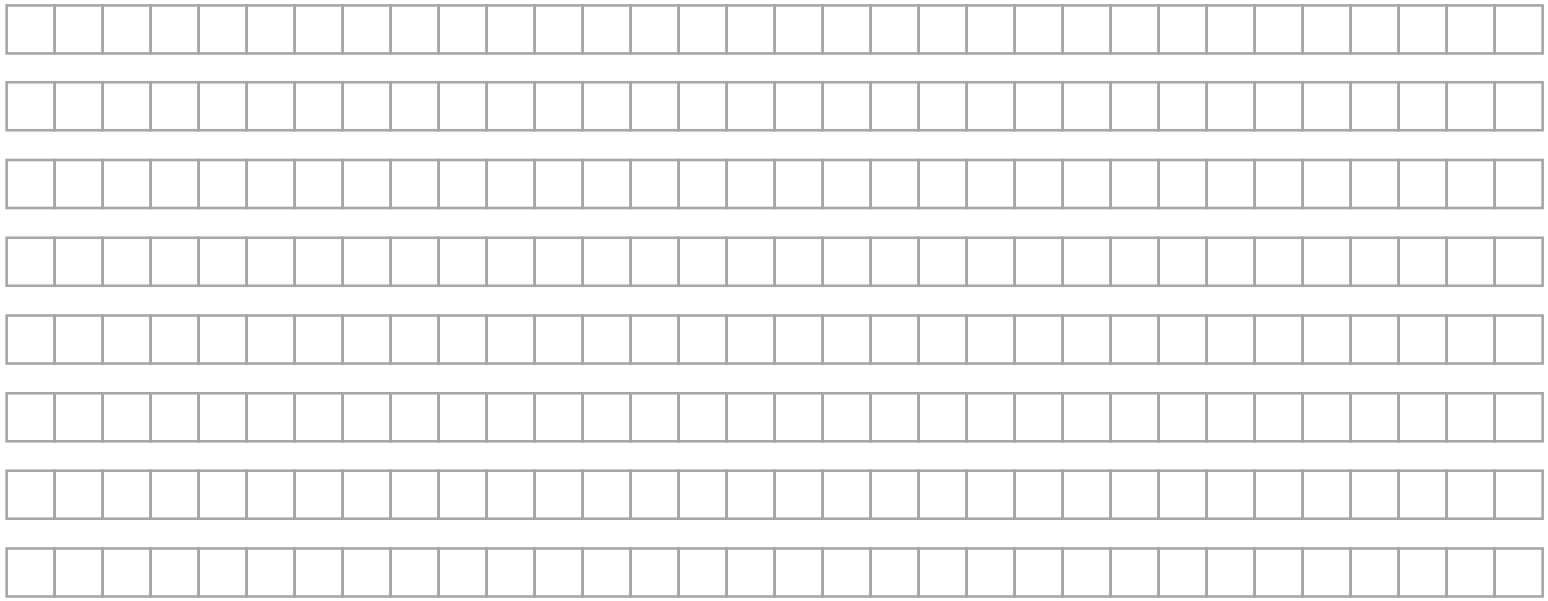
Suppose that you are going to lay out the data storage for the relation shown at right. Assume that the disk blocks are just 32 bytes long, as represented by the lines at the bottom of this page. How would the data be stored, given the following policies?

- Tuple headers contain just the total record length, stored as a single byte. No schema pointer or timestamp.
- Variable-length fields appear after all the fixed-length fields. Each (except the last) begins with one byte giving the length of the data portion.
- There are no spanned records. Extra space at the end of a block is left unused, and filled with zeros.

CREATE TABLE DogLicenses(name VARCHAR, licenseNo INTEGER, rabiesVac BOOLEAN, expires CHAR(3));

Name	licenseNo	rabiesVac	expires
Adiko	1001	Y	OCT
Butch	1002	Y	JAN
Caro	1003	Y	MAR
Daiki	1004	Y	MAR
Eduardo	1005	Y	DEC
Fido	1006	N	AUG
Rover	1007	Y	FEB
Spuds	1008	N	JUL
Tiifu	1009	Y	SEP

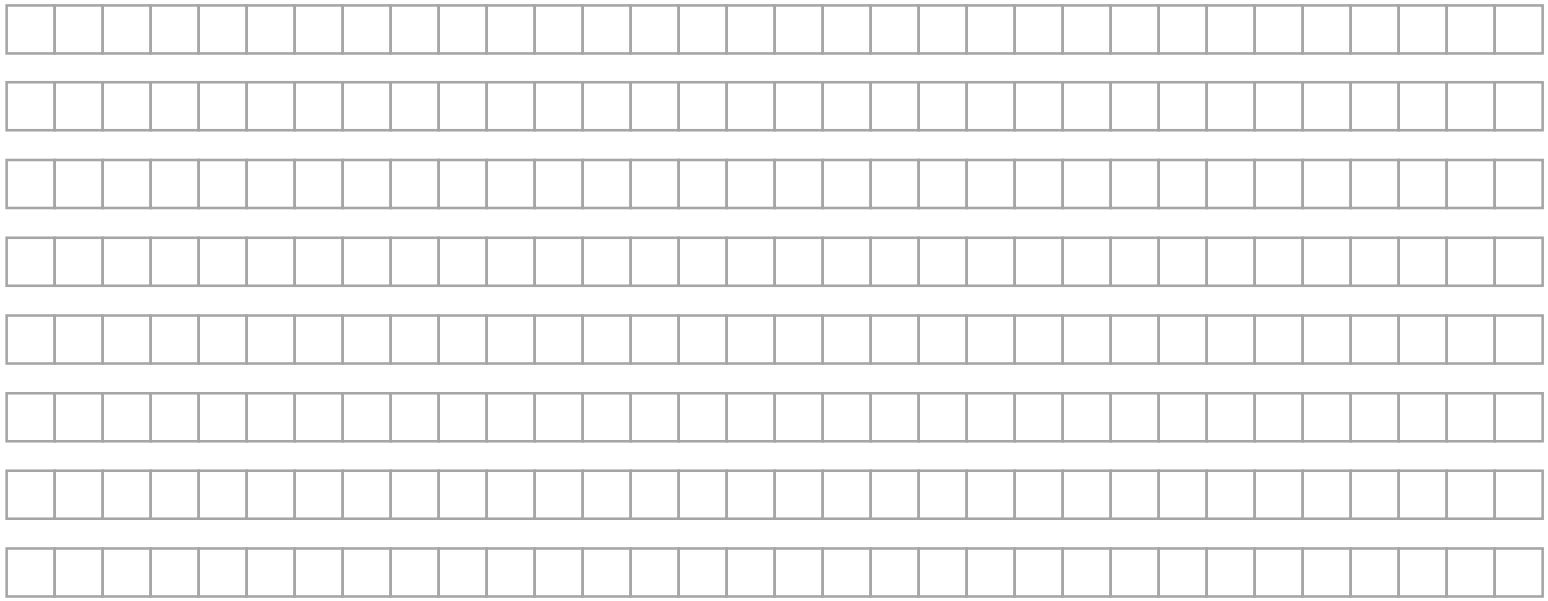


Suppose that you are going to lay out the data storage for the relation shown at right. Assume that the disk blocks are just 32 bytes long, as represented by the lines at the bottom of this page. How would the data be stored, given the following policies?

- There are no tuple headers. Assume the schema is known, and thus the record length.
- Variable-length fields are stored in separate blocks. Use one byte to give the block number, one for the position.
- For blocks of variable data, use an offset table.
- There are no spanned records. Extra space at the end of a block is left unused, and filled with zeros.

CREATE TABLE DogLicenses(name VARCHAR, licenseNo INTEGER, rabiesVac BOOLEAN, expires CHAR(3));

Name	licenseNo	rabiesVac	expires
Adiko	1001	Y	OCT
Butch	1002	Y	JAN
Caro	1003	Y	MAR
Daiki	1004	Y	MAR
Eduardo	1005	Y	DEC
Fido	1006	N	AUG
Rover	1007	Y	FEB
Spuds	1008	N	JUL
Tiifu	1009	Y	SEP



Suppose that you are trying to read the relation shown at right. Assume that the disk blocks are just 32 bytes long, as shown by the lines at the bottom of this page. How would you interpret the data below, given the following policies?

- Tuple headers contain just the total record length, stored as a single byte. No schema pointer or timestamp.
- Variable-length fields appear after all the fixed-length fields. Each (except the last) begins with one byte giving the length of the data portion.
- There are no spanned records. Extra space at the end of a block is left unused, and filled with zeros.

CREATE TABLE DogLicenses(name VARCHAR, licenseNo INTEGER, rabiesVac BOOLEAN, expires CHAR(3));

name	licenseNo	rabiesVac	expires

15	1	0	0	1	Y	O	C	T	5	A	d	i	k	o	15	1	0	0	2	Y	J	A	N	5	B	u	t	c	h	0	0
18	1	0	0	3	Y	M	A	R	8	C	o	n	c	h	i	t	a	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	1	0	0	4	Y	M	A	R	5	D	a	i	k	i	17	1	0	0	5	Y	D	E	C	7	E	d	u	a	r	d	o
14	1	0	0	6	N	A	U	G	4	F	i	d	o	13	1	0	0	7	Y	F	E	B	3	R	e	y	0	0	0	0	0
14	1	0	0	8	N	J	U	L	4	S	p	o	t	15	1	0	0	9	Y	S	E	P	5	T	i	i	f	u	0	0	0

Suppose that you are trying to read the relation shown at right. Assume that the disk blocks are just 32 bytes long, as shown by the lines at the bottom of this page. How would you interpret the data below, given the following policies?

- There are no tuple headers. Assume the schema is known, and thus the record length.
- Variable-length fields are stored in separate blocks. Use one byte to give the block number, one for the position.
- For blocks of variable data, use an offset table.
- There are no spanned records. Extra space at the end of a block is left unused, and filled with zeros.

CREATE TABLE DogLicenses(name VARCHAR, licenseNo INTEGER, rabiesVac BOOLEAN, expires CHAR(3));

Name	licenseNo	rabiesVac	expires

1	0	1	0	0	1	Y	O	C	T	1	1	1	0	0	2	Y	J	A	N	1	2	1	0	0	3	Y	M	A	R	0	0
27	22	18	13	6	0	E	d	u	a	r	d	o	D	a	i	k	i	C	a	r	o	B	u	t	c	h	A	d	i	k	0
1	3	1	0	0	4	Y	M	A	R	1	4	1	0	0	5	Y	D	E	C	3	0	1	0	0	6	N	A	U	G	0	0
28	23	18	13	0	0	0	0	0	0	0	0	0	T	i	i	f	u	S	p	u	d	s	R	o	v	e	r	F	i	d	o
3	1	1	0	0	7	Y	F	E	B	3	2	1	0	0	8	N	J	U	L	3	3	1	0	0	9	Y	S	E	P	0	0

Suppose that you are going to lay out the data storage for the relation shown at right. Assume that the disk blocks are just 32 bytes long, as represented by the lines at the bottom of this page. How would the data be stored, given the following policies?

- Tuple headers contain just the total record length, stored as a single byte. No schema pointer or timestamp.
- Variable-length fields appear after all the fixed-length fields. Each (except the last) begins with one byte giving the length of the data portion.
- There are no spanned records. Extra space at the end of a block is left unused, and filled with zeros.

CREATE TABLE DogLicenses(name VARCHAR, licenseNo INTEGER, rabiesVac BOOLEAN, expires CHAR(3));

name	licenseNo	rabiesVac	expires
Adiko	1001	Y	OCT
Butch	1002	Y	JAN
Conchita	1003	Y	MAR
Daiki	1004	Y	MAR
Eduardo	1005	Y	DEC
Fido	1006	N	AUG
Rey	1007	Y	FEB
Spot	1008	N	JUL
Tiifu	1009	Y	SEP

15	1	0	0	1	Y	O	C	T	5	A	d	i	k	o	15	1	0	0	2	Y	J	A	N	5	B	u	t	c	h	0	0
18	1	0	0	3	Y	M	A	R	8	C	o	n	c	h	i	t	a	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	1	0	0	4	Y	M	A	R	5	D	a	i	k	i	17	1	0	0	5	Y	D	E	C	7	E	d	u	a	r	d	o
14	1	0	0	6	N	A	U	G	4	F	i	d	o	13	1	0	0	7	Y	F	E	B	3	R	e	y	0	0	0	0	0
14	1	0	0	8	N	J	U	L	4	S	p	o	t	15	1	0	0	9	Y	S	E	P	5	T	i	i	f	u	0	0	0

Suppose that you are going to lay out the data storage for the relation shown at right. Assume that the disk blocks are just 32 bytes long, as represented by the lines at the bottom of this page. How would the data be stored, given the following policies?

- There are no tuple headers. Assume the schema is known, and thus the record length.
- Variable-length fields are stored in separate blocks. Use one byte to give the block number, one for the position.
- For blocks of variable data, use an offset table.
- There are no spanned records. Extra space at the end of a block is left unused, and filled with zeros.

CREATE TABLE DogLicenses(name VARCHAR, licenseNo INTEGER, rabiesVac BOOLEAN, expires CHAR(3));

Name	licenseNo	rabiesVac	expires
Adiko	1001	Y	OCT
Butch	1002	Y	JAN
Caro	1003	Y	MAR
Daiki	1004	Y	MAR
Eduardo	1005	Y	DEC
Fido	1006	N	AUG
Rover	1007	Y	FEB
Spuds	1008	N	JUL
Tiifu	1009	Y	SEP

1	0	1	0	0	1	Y	O	C	T	1	1	1	0	0	2	Y	J	A	N	1	2	1	0	0	3	Y	M	A	R	0	0
27	22	18	13	6	0	E	d	u	a	r	d	o	D	a	i	k	i	C	a	r	o	B	u	t	c	h	A	d	i	k	0
1	3	1	0	0	4	Y	M	A	R	1	4	1	0	0	5	Y	D	E	C	3	0	1	0	0	6	N	A	U	G	0	0
28	23	18	13	0	0	0	0	0	0	0	0	0	T	i	i	f	u	S	p	u	d	s	R	o	v	e	r	F	i	d	o
3	1	1	0	0	7	Y	F	E	B	3	2	1	0	0	8	N	J	U	L	3	3	1	0	0	9	Y	S	E	P	0	0